RESTORATION OF PULSE POSITION MODULATED SIGNAL IN A HIGH NOISE ENVIRONMENT

THESIS

AFIT/GE/EE/82D-31 Kenneth N. Frankovich
Captain USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

AFIT/GE/EE/82D-31

RESTORATION OF PULSE POSITION MODULATED SIGNAL IN A HIGH NOISE ENVIRONMENT

THESIS

AFIT/GE/EE/82D-31 Kenneth N. Frankovich
Captain USAF

E

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/82D-31	2. GOVT ACCESSION NO. AD-A124	3. RECIPIENT'S CATALOG NUMBER 902
4. TITLE (and Subtitle) Restoration of Pulse Position Modulated Signal in a High Noise Environment		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Kenneth N. Frankovich, Captain, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology, AFIT/EN Wright-Patterson AFB, Dayton, OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1982 13. NUMBER OF PAGES 253
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

[illegible] AFR 190-17.

Lynn E. Wolaver

LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

4 JAN 1983

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Amplitude Estimation
Markov Process
Anamolous Pulse Error
Viterbi Algorithm
Maximum a Posteriori
Finite State Sequence

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The performance of a pulse position modulation (PPM) receiver, operating at a predetection signal-to-noise ratio below threshold, is severely degraded by the occurrence of anamolous pulses. The anamolies are caused by noise with sufficient amplitude to be detected as a signal pulse. The resultant multiple threshold crossings per sample frame severely degrade the demodulated version of the original message.

(continued on reverse)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block #20. (Cont'd)

In this report, a maximum a posteriori (MAP) amplitude sequence estimator is developed based on the Viterbi algorithm. The signal is modelled as a finite-state, discrete-time Markov process. The states correspond to the possible positions within a quantized version of the sample frame. A receiver structure and associated operating characteristics are presented in support of the concept of a quantized PPM frame. The pulse position/amplitude sequence is observed in white Gaussian noise, and probability of detection for each signal pulse is assumed equal to one.

Monte Carlo simulation is used to determine the average performance of the estimator. The performance is expressed in terms of ensemble average mean square error, ensemble average state estimate error and frequency of error. The frequency of error is termed the average probability of error. The average probability is expressed with respect to the total average number of threshold crossings.

Each frame is quantized into 32 pulse positions. The performance of the estimator is presented for a family of six signals. The signal set was selected arbitrarily with the only requirement for consideration being that the signal had to consist of at least one absolute maximum during the observation time.

The estimator is shown to be able to provide an accurate reconstruction of the original pulse position/amplitude sequence for the signals considered for up to 20 or 23 pulse occurrences per frame, depending on the signal. The ability to enhance estimator performance by varying various signal model and algorithm parameters is also presented. The algorithm is shown to be sensitive to the assumed model (model sensitivity) but not to the set of transition probability values (rate sensitivity) within a given model.

AFIT/GE/EE/82D-31

RESTORATION OF PULSE POSITION MODULATED SIGNAL IN A HIGH NOISE ENVIRONMENT

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Kenneth N. Frankovich, B.S.E.E.
Captain USAF
Graduate Electrical Engineering

December 1982

Approved for public release; distribution unlimited.

ACKNOWLEDGEMENTS

I would like to extend my thanks to my thesis advisor, Lieutenant Colonel Ronald J. Carpinella, PhD, MSEE, BSEE, for his guidance assistance and thought-provoking discussions concerning this project over the last nine months; and to my committee members, Major Kenneth G. Castor, MSEE, BSEE, and Professor Matthew Kabrisky, for their comments and suggestions. A special thank you is also due to Major Larry L, Kizer, MSEE, BSEE, for allowing the Eclipse computer to be used for other than speech or visual pattern processing.

Kenneth N. Frankovich

PREFACE

This study was prompted by a pulse position/amplitude estimation problem of a pulse position modulation receiver operating well below threshold. The receiver suffers from extensive anamolous pulse-position errors while guaranteeing a minimum probability of detection for the signal pulse.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The performance of a pulse position modulation (PPM) receiver, operating at a predetection signal-to-noise ratio below threshold, is severely degraded by the occurrence of anamolous pulses. The anamolies are caused by noise with sufficient amplitude to be detected as a signal pulse. The resultant multiple threshold crossings per sample frame severely degrade the demodulated version of the original message.

In this report, a maximum a posteriori (MAP) amplitude sequence estimator is developed based on the Viterbi algorithm. The signal is modelled as a finite-state, discrete-time Markov process. The states correspond to the possible positions within a quantized version of the sample frame. receiver structure and associated operating characteristics are presented in support of the concept of a quantized PPM frame. The pulse position/amplitude sequence is observed in white Gaussian noise, and probability of detection for each signal pulse is assumed equal to one.

Monte Carlo simulation is used to determine the average performance of the estimator. The performance is expressed in terms of ensemble average mean square error, ensemble average state estimate error and frequency of error. The frequency of error is termed the average probability of error.

The average probability is expressed with respect to the total average number of threshold crossings.

Each frame is quantized into 32 pulse positions. The performance of the estimator is presented for a family of six signals. The signal set was selected arbitrarily with the only requirement for consideration being that the signal had to consist of at least one absolute maximum during the observation time.

The estimator is shown to be able to provide an accurate reconstruction of the original pulse position/amplitude sequence for the signals considered for up to 20 or 23 pulse occurrences per frame, depending on the signal. The ability to enhance estimator performance by varying various signal model and algorithm parameters is also presented. The algorithm is shown to be sensitive to the assumed model (model sensitivity) but not to the set of transition probability values (rate sensitivity) within a given model.

I. INTRODUCTION:

Pulse position modulation (PPM) is one of three analog pulse modulation techniques; the other two being pulse-amplitude modulation (PAM) and pulse-width modulation (PWM). For PAM, a message m(t) is sampled and a pulse, with amplitude proportional to the sample value of the message, is transmitted across the channel. In both PWM and PPM, the information resides in the position of the transmitted pulse instead of its amplitude. Demodulation of PWM or PPM is accomplished by detecting the edge (leading or trailing) of the transmitted pulse. The performance of PAM is essentially equivalent to continuous baseband transmission (Ref 20). Both PWM and PPM are nonlinear modulation techniques and, therefore, performance is appropriately expressed in terms of the signal-to-noise ratio (SNR) - bandwidth (BW) trade-off and threshold effect experienced by nonlinear modulation techniques.

For PPM, the instantaneous amplitude sample of the message m(t), is caused to vary the position in time of a pulse relative to its unmodulated time of occurrence. Compared to PWM, PPM conserves signal power. A PPM pulse has fixed amplitude and fixed duration and, therefore, the energy in the transmitted pulse is independent of the sample value of m(t). PPM is generally preferred over PWM primarily for this reason and is generally used when bandwidth is not at a premium for such applications as multiplexing telephone channels or transmission of some types of guidance and/or telemetry data.

A PPM receiver generally uses a threshold method of detection and demodulation. The receiver threshold is set to detect an expected pulse amplitude within a sample or frame time. The occurrence time of an unmodulated pulse is frequently referred to as the frame marker. The occurrence time of a modulated pulse is determined by the time the modulated pulse is detected (crossing the threshold). The time is measured with respect to the frame marker. The demodulation of a PPM signal is simply a mapping of occurrence times into amplitudes. In the absence of noise, the receiver maps the time of the threshold crossing into a unique amplitude value. PPM can be thought of as a discrete time, continuous amplitude modulation technique. Figure 1 depicts an ideal, noise free PPM signal for a single frame.

A characteristic of nonlinear modulation techniques such as PPM, is the ability to trade-off bandwidth and SNR to enhance performance. As with all nonlinear modulation techniques, PPM is vulnerable to threshold effect. The point at which the post-detection SNR falls off rapidly with decreasing predetection SNR is called the threshold (not to be confused with the detection threshold). All nonlinear modulation techniques experience this threshold. A

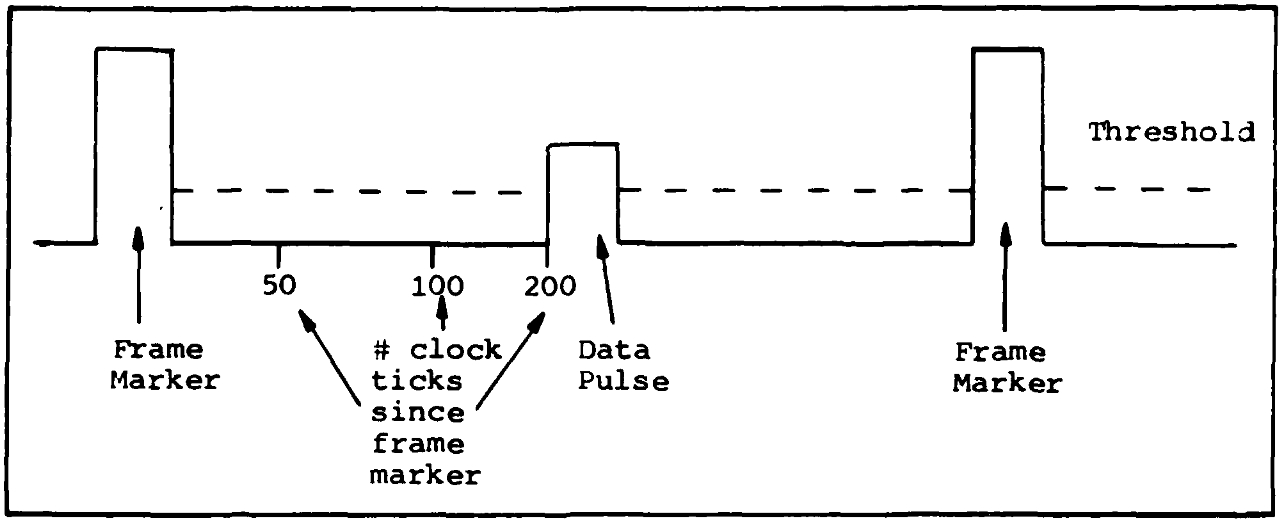


FIGURE 1. Ideal PPM Signal

predetection SNR below the threshold causes the receiver to "break," or lose the signal. The resulting severely degraded performance is due to this threshold effect. It is possible to make trade-offs between SNR and BW to maintain acceptable receiver performance as long as the receiver remains above the threshold value of predetection SNR.

Errors in the demodulation of PPM signals occur in two basic forms. The least critical demodulation error occurs when the receiver operates above threshold. This type of error is a result of the channel noise, $n(t_i)$, causing a net incremental shift in the occurrence time of the modulated pulse as shown in Figure 2. The noise causes the receiver to detect the modulated pulse slightly early (or late) with respect to the no-noise position, $e(t_i)$ in Figure 2. The error becomes more significant with decreasing

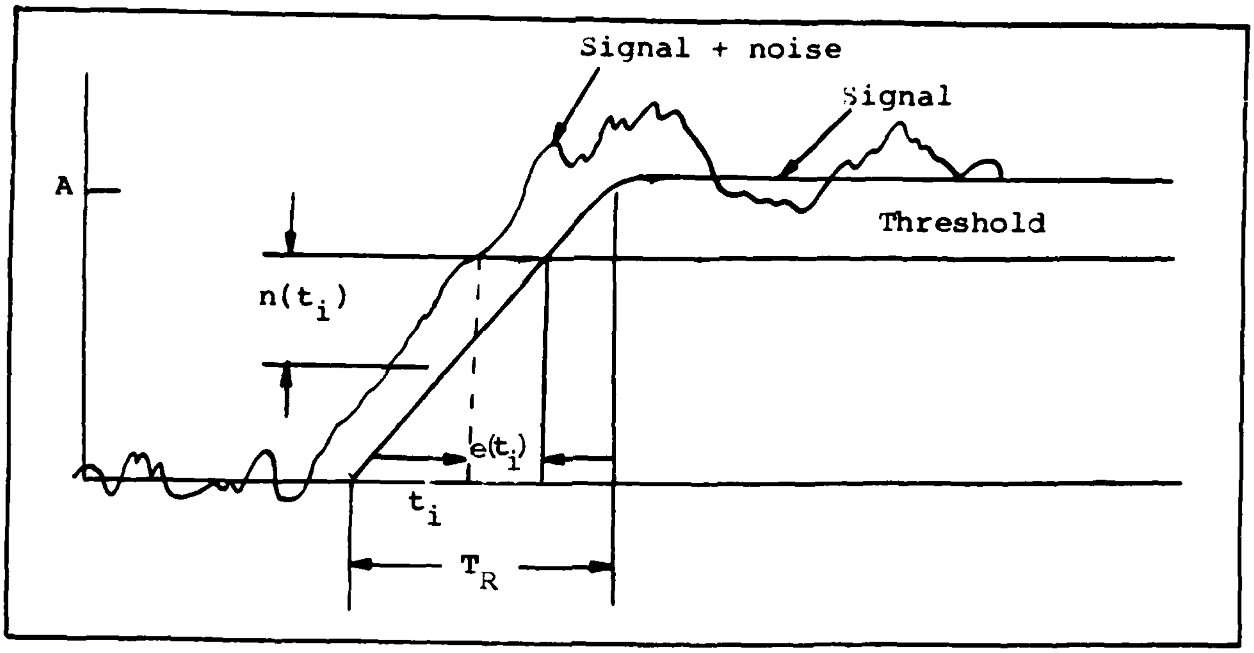


FIGURE 2. PPM Signal With Noise

predetection SNR. For low SNR, this error may be minimized by using a wider system bandwidth to allow for pulses with faster rise time (smaller T_R in Figure 2). A sharper pulse reduces the effect the noise has in producing the incremental shift error. The SNR-BW trade-off is effective only to a point. The combination of low predetection SNR and wide bandwidth eventually forces the receiver to operate below threshold. Threshold effect in PPM produces the second, and more significant demodulation error to be referred to as anamolous detection error.

An anamolous error is synonymous with false threshold crossings. The false threshold crossings result when noise spikes occur with sufficient amplitude to cross the receiver threshold. The receiver detects and demodulates multiple "pulses" within each frame instead of the desired single data pulse. The erroneous threshold crossings result in significant demodulation (pulse position) error. Figure 3 shows an example of anamolous errors within a single frame.

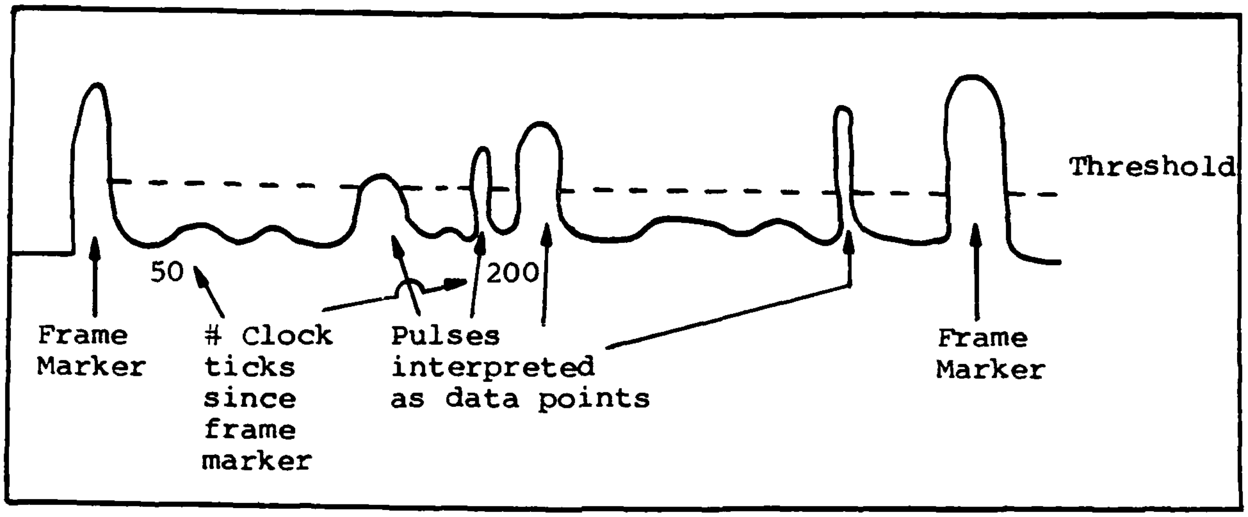

Threshold
50
200
Frame Marker
# Clock ticks since frame marker
Pulses interpreted as data points
Frame Marker

FIGURE 3. Received Signal After Transmission Through Noisy Medium With Multiple Anamolous Errors

BACKGROUND:

The PPM system under study essentially operates close to or below threshold. Anamolous errors are caused by the low SNR, wide bandwidth combination. The extent of anamolous error is exacerbated, however, by the need to guarantee detection of the actual data pulse within each frame. The detection threshold is set low enough to provide a probability of detection, P_D=1.0 for the actual data pulse. The number of anamolies is the result of the low predetection SNR, wide BW, and low detection threshold.

A fix designed to minimize the effect of the anamolous errors is to place the receiver in a mode in which the first threshold crossing within a frame constitutes the pulse occurrence time and all other threshold crossings within that frame are ignored (Ref 15). This mode is referred to as frame lock-out. The subject receiver instead, detects and demodulates all threshold crossings within each frame. Unlike the frame lock-out method, this mode of operation provides the desired probability of detection assuming the detection threshold is set appropriately.

A system that either does not suffer from anamolous errors or that employs the aforementioned fix will map the demodulated pulse position into a single amplitude value. A typical mapping is given in Figure 4. The given PPM system maps each threshold crossing into an amplitude value. This results in an output consisting of multiple amplitude values versus discrete sample times as seen in Figure 5. It is obvious that some form of reliable single point data is desirable over the resultant multipoint data.

AMPLITUDE ESTIMATION PROBLEM:

As described previously, the combination of low SNR, wide bandwidth and receiver mode of operation adversely affects the task of amplitude estimation (single point amplitude mapping of the detected threshold crossings). The basic problem is the reconstruction of the original

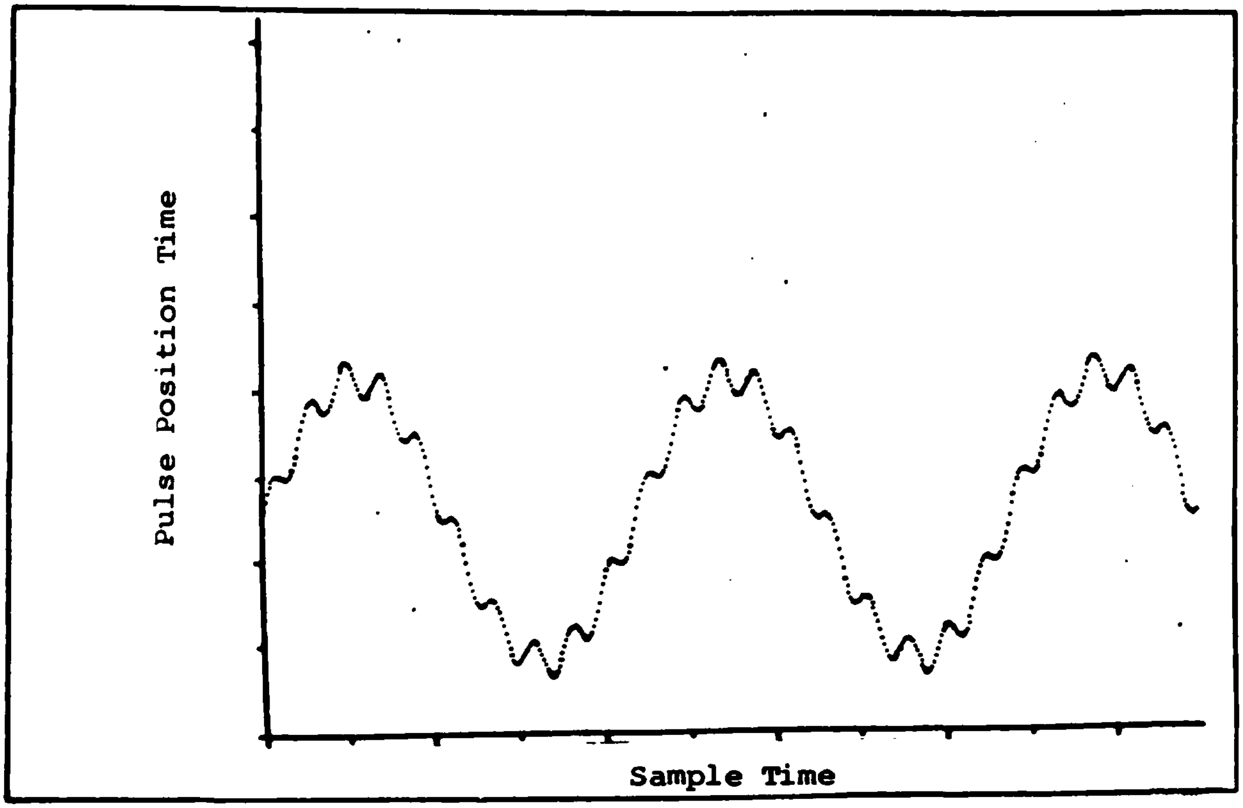


FIGURE 4. An Ideal Signal as it Leaves the Transmitter

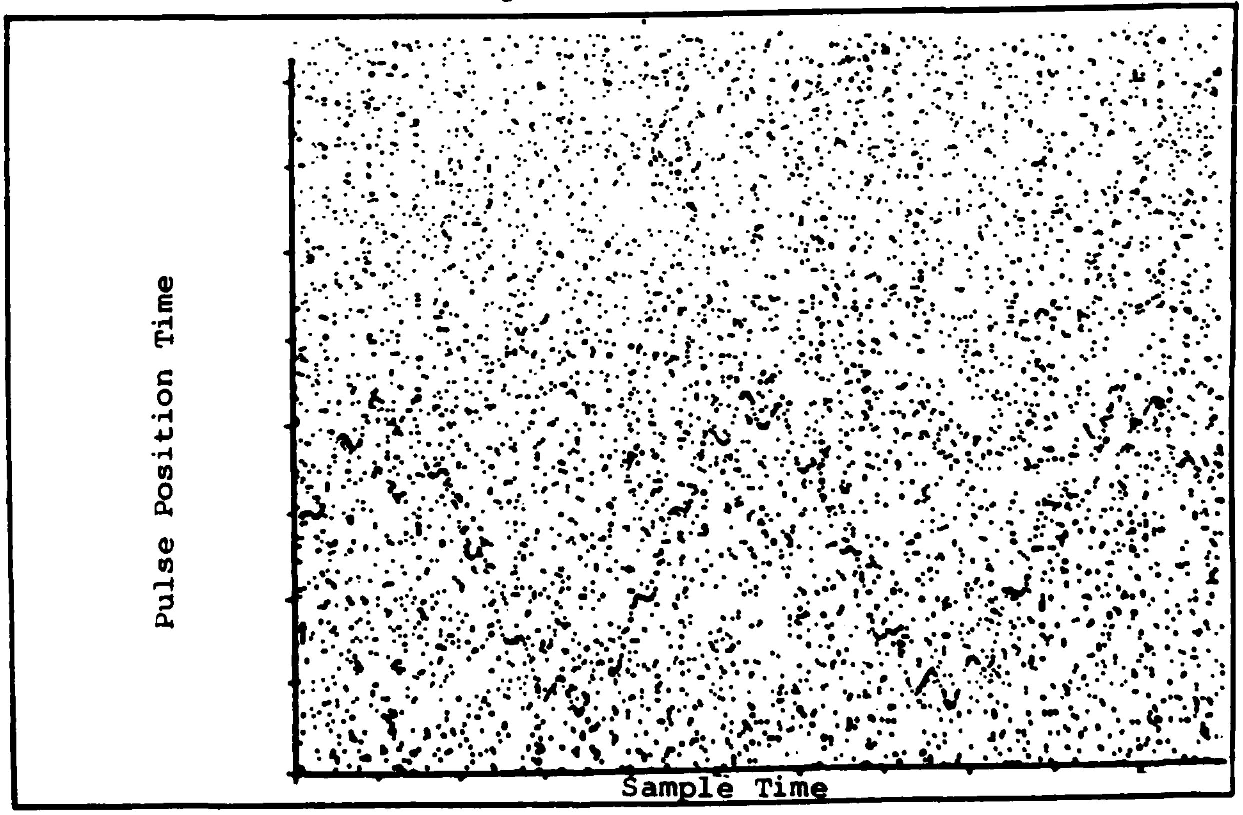


FIGURE 5. Received Noisy PPM Signal With Multiple Anamolous Errors

message. Assuming the receiver mode cannot be changed, an automated signal processing technique is desired to convert the multipoint PPM data to single point data with a high level of confidence that the edited data accurately reconstructs the original message signal.

The problem will be approached as a pulse position/ amplitude estimation problem. The amplitude estimator utilized in this thesis is one which produces the maximum a posteriori (MAP) estimate of a signal amplitude sequence. Actual computation of the MAP amplitude sequence employs the Viterbi Decoding Algorithm (backward dynamic programming) (Ref 18).

SCOPE AND ASSUMPTIONS:

In this thesis, the amplitude sequence of the message signal will be modelled as a first order Markov process. A special case discrete time Markov process will be presented as a suitable signal model. The channel is assumed to be memoryless and additive. The Viterbi algorithm, therefore, provides a MAP estimate if and only if Markov signal and memoryless channel constraints are met. Both Guassian and Poisson channel models will be considered. Amplitude estimation will be made on a sample-by-sample basis based on quantized versions of an assumed set of signals. The significance of providing a quantized estimate will be presented later.

Actual signal power and noise power are unavailable and, therefore, system performance will be expressed in terms of probability of estimator error (to be described later), ensemble average estimator error, and ensemble average mean-square error with respect to a zero crossing ratio (also to be defined later).

Assuming a receiver operating mode that produces numerous anamolies per frame, exact message reconstruction is assumed unattainable. Additionally, if erroneous threshold crossings do not exist (i.e., operation well above threshold), conventional PPM detection and demodulation are sufficient with demodulation errors attributable to incremental shift phenomena only. This thesis is concerned with presenting a signal model, a channel model(s), and an automated estimation technique that together will provide a reasonable reconstruction of a transmitted message that has been distorted by the channel and receiver mode as described. This thesis does not attempt to provide an all encompassing improvement in methodology in detecting and demodulating PPM signals. The automation process (MAP estimator) presented in this thesis is presented as an alternative to the frame lock-out technique in reducing the affect these anamolies have on demodulation. This approach differs from the upcrossing receiver (Ref 13:307) in that the upcrossing receiver assumes a high SNR and low probability of anamoly occurrence. The upcrossing receiver is a maximum likelihood receiver designed to minimize incremental shift errors experienced when operating above threshold and no large anamolous errors exist.

Lastly, the limits on the application of the estimator will be identified. A limiting point will be identified, at which given the signal model parameters, Viterbi algorithm parameters, and the assumed message set, a high degree of confidence in signal reconstruction is no longer attainable.

OVERVIEW:

This report begins with a description of the underlying system constraints and the nature of the amplitude estimation problem. Chapter II presents the receiver structure and operating characteristics necessary to allow for this approach to the problem. The signal amplitude sequence model, a Markov process model, is presented as well as the algorithm for a MAP amplitude sequence estimator using the Viterbi algorithm. A discussion of the algorithm simulation and the results achieved are presented in Chapter III. Finally, conclusions are reviewed and recommendations for further study are presented in Chapter IV.

II. THEORETICAL BACKGROUND:

In this chapter, the theoretical background of the MAP amplitude sequence estimator is presented. The chapter is divided into three sections. The first section presents the receiver structure and operating characteristics. The second section provides a brief discussion on Markov processes, and presents the Markov model used for the discrete representation of the signal. The amplitude sequence estimation is performed via the Viterbi algorithm. The third section of the chapter presents an expression (path metric) to be used recursively to solve the estimation problem and a discussion of the application of the algorithm to this estimation problem.

RECEIVER STRUCTURE:

A conventional PPM system detects the occurrence of a pulse during a specified frame or sample time by threshold crossing detection. The receiver may be designed to detect the leading edge, trailing edge, or peak of the pulse within each frame time. The frame is "marked" by two stationary pulses called frame markers. The markers are the unmodulated pulse positions in time. The receiver threshold, K, is generally set at one half the expected pulse amplitude (Ref 15). A typical frame is depicted in Figure 6.

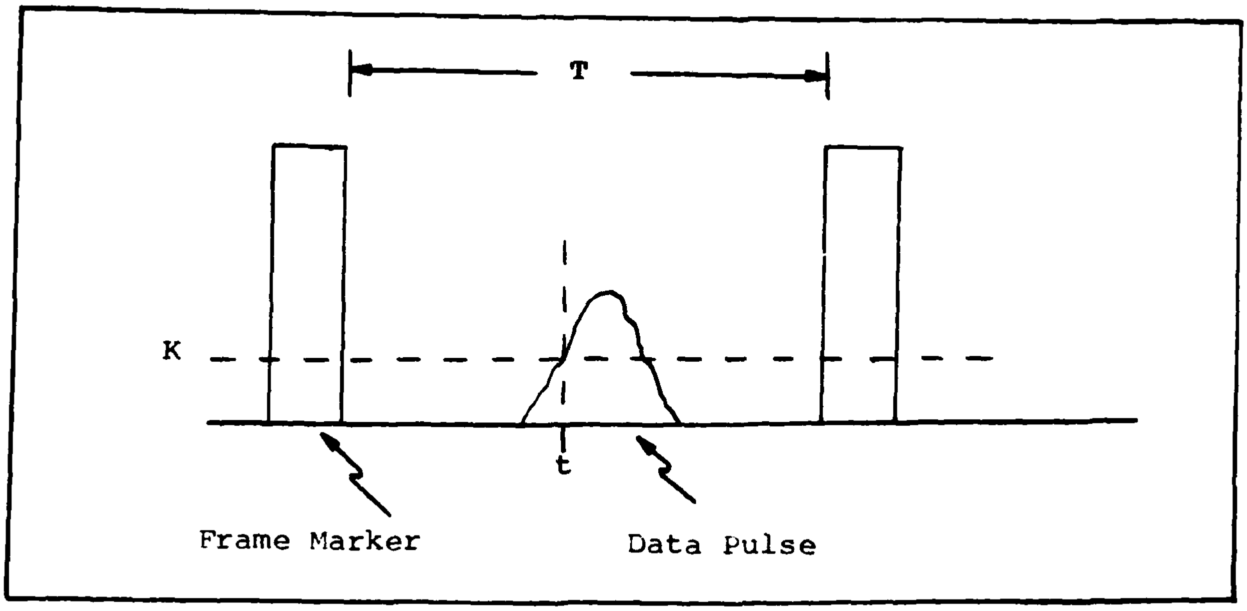


FIGURE 6. Detection of Modulated Pulse

As depicted, the time t, at which the receiver detects the threshold crossing, is demodulated or mapped into a unique amplitude value. The range of amplitude values depends upon the frame time. It is desirable to represent this continuum of amplitudes in a discrete/quantized form, which in turn, reflects a quantized version of each frame time. Reasons for this quantization will be presented in the next section.

It is necessary to propose a new receiver structure based on quantized signal amplitudes and, therefore, also quantized pulse positions. The frame time T, is divided into M uniform intervals of time. These M intervals represent M possible pulse positions. Consequently, the M intervals also represent M possible signal amplitudes. The demodulator mapping is now one of discrete amplitude versus discrete sample time. The M intervals are independent observation intervals or "bit times." A threshold crossing within a bit time is represented by a "1" and no threshold crossing within a bit time is represented by a "0." Figure 7 shows an example of a sample frame subdivided into M=16, pulse position intervals. These 16 possible pulse positions are mapped by the demodulation into 16 amplitude levels. The 16 intervals define a 16 bit word. The word in Figure 7 would be 0000001000000000.

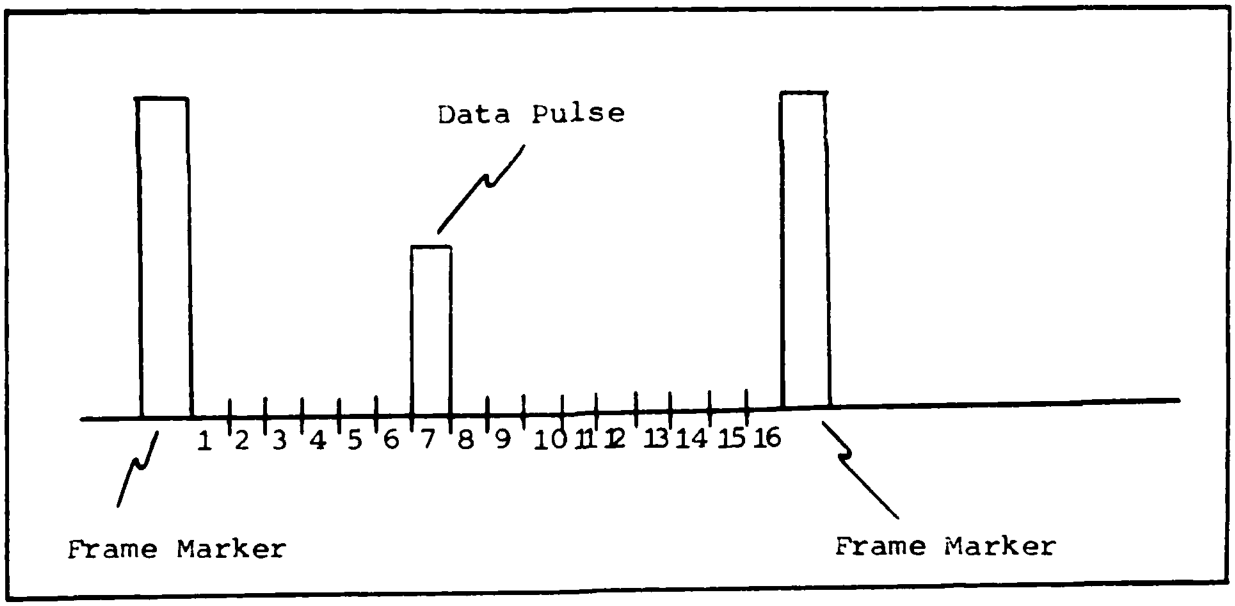


FIGURE 7. Quantized Ideal PPM Frame

If natural sampling is assumed, the jth frame represents the jth signal sample. The jth frame may then be represented as an M-dimensional vector:

$$\underline{Z}_j = \begin{bmatrix} Z_i \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ Z_M \end{bmatrix} \tag{1}$$

where each element Z_i represents the threshold crossings (0 or 1) for the i^{th} bit time. If the sampling is further assumed to be uniform, each frame represents one of N, independent samples. The receiver may now be modelled as processing a sequence of N, M-dimensional, independent vectors. Each vector $\underline{Z}_j$ is defined as an observation vector.

Using the conventional notation for an additive channel, the system may be represented in vector form as:

$$\underline{Z} = \underline{S} + \underline{n} \tag{2}$$

where $\underline{Z}$ is the observation vector, $\underline{S}$ is some signal vector, and $\underline{n}$ represents the noise due to the channel. The signal vector must now be defined.

A signal vector is defined as an M-dimensional vector

$$\underline{S}^i = \begin{bmatrix} S_K \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ S_M \end{bmatrix} \tag{3}$$

composed of all zero elements except for a single element:

$$S_K = 1 \tag{4}$$

corresponding to a pulse in the K^{th} position. The message m(t), may now be expressed as sequence of N, M-dimensional signal vectors:

$$m(t) = \sum_{j=1}^{N} \underline{S}_j \, \Phi(t-jT) \tag{5}$$

where each $\underline{S}_j^i$ is an element of the set of M possible signal vectors corresponding to the M possible pulse positions. A more complete form of equation (5) is:

$$m(t) = \sum_{j=1}^{N} \underline{S}_j^i \, \Phi(t-jT) \qquad i = 1, 2, \ldots. M \tag{6}$$

where $\Phi(t)$ is the orthogonal basis set with dimension M formed by the M possible pulse positions. The signal set S is defined as:

$$S = \left[\underline{S}^1, \underline{S}^2, \ldots . \underline{S}^M\right] \tag{7}$$

where:

$$\underline{S}^1 = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad \underline{S}^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad \ldots\ldots \quad \underline{S}^M = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} \tag{8}$$

Before defining the noise vector $\underline{n}$, the receiver operating characteristics must be highlighted.

The receiver operates in what must be called a "less than optimum" manner. Factors affecting receiver performance are bandwidth, SNR, and the threshold setting. The bandwidth is excessively wide in order to allow for a wide range of signal pulse rise times. The predetection SNR is known to be very low (less than OdB). The threshold setting is not fixed in this receiver. The threshold is set to quarantee detection of the signal pulse with the probability of detection, $P_D = 1.0$ for all frame times. The low SNR, wide bandwidth, and low threshold setting places the receiver in a regime where the largest demodulation error is due to erroneous threshold crossings caused by the noise (Ref 13, 15, 19).

In terms of classical receiver operating characteristics (ROC), the region of operation may be seen graphically on a family of ROC curves (Ref 7). The ROC curves of Figure 8 represent general receiver operating characteristics on a bit time basis. In Figure 8, P_{FA} is the probability of false alarm or the probability that in a single bit time, a threshold crossing will occur given that the signal pulse was not sent in that interval. The range of P_{FA} for each of the ROC curves is depicted by the line segments AX, BX, CX, and DX in Figure 8. Again, on bit time basis, the system may be represented in terms of a discrete channel model with bit crossover probabilities as depicted in Figure 9. In Figure 9, X_i denotes the i^{th} bit time in a frame that is transmitted, and Z_i the i^{th} observed bit time. The transition probabilities $P(Z_i|X_i)$ are simply:

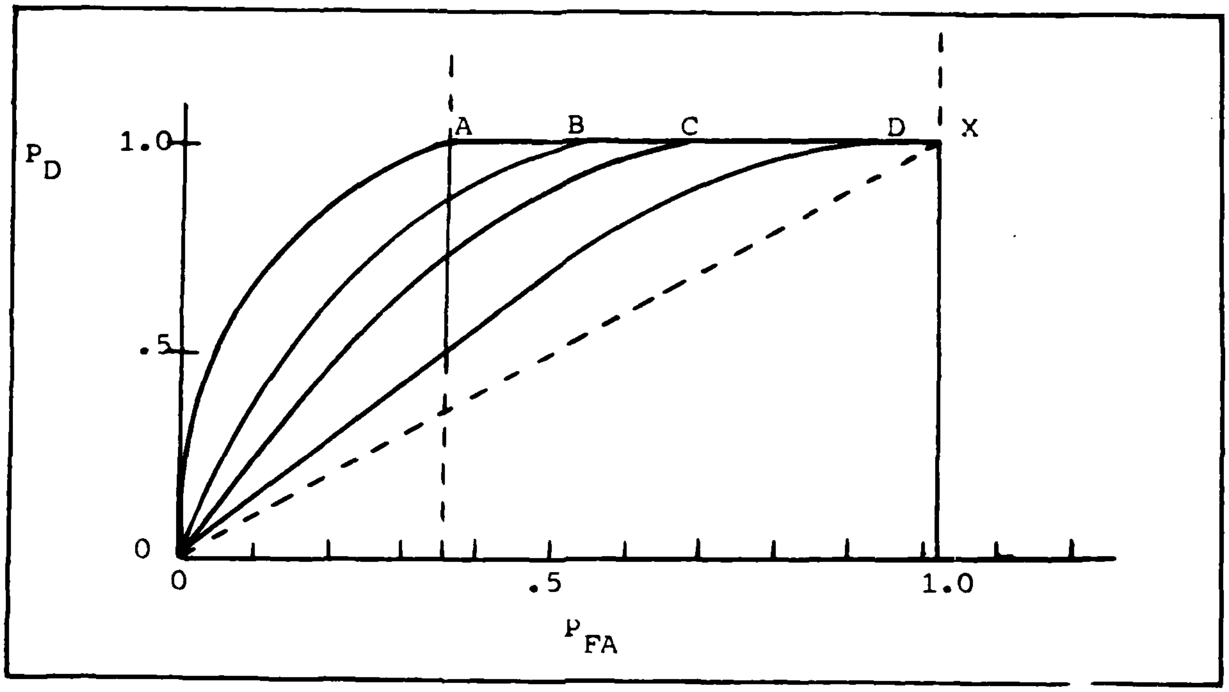


FIGURE 8. General ROC Curves

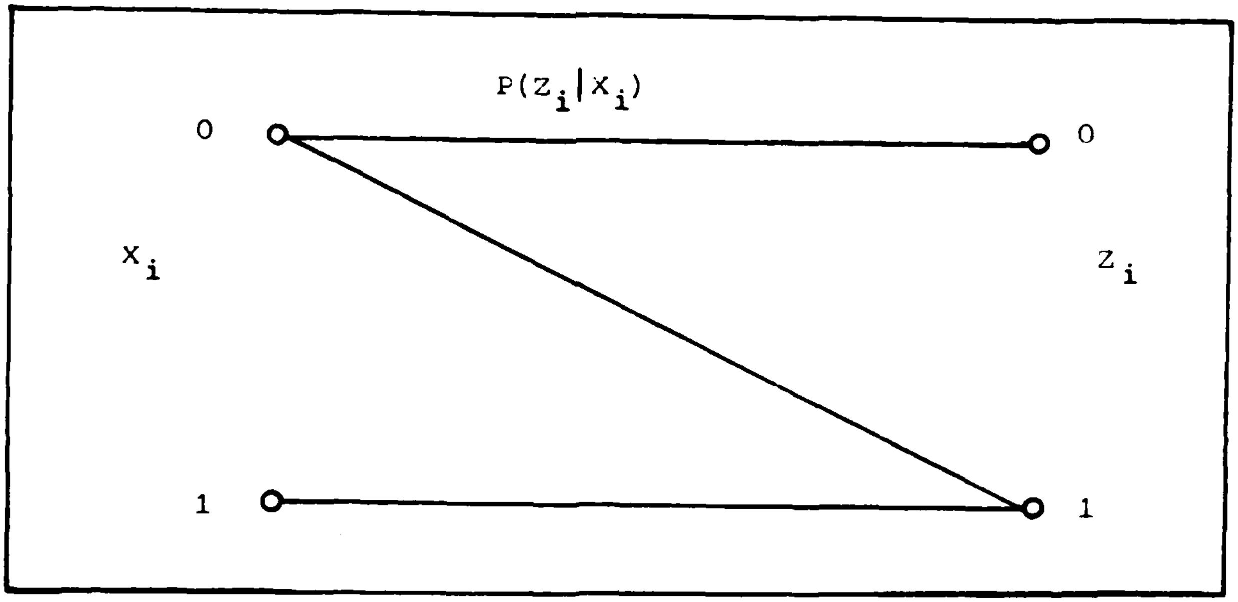


FIGURE 9. Discrete Channel Model

$$P(1|1) = P_D = 1.0$$
$$P(1|0) = P_{FA} \tag{9}$$
$$P(0|0) = 1 - P_{FA}$$
$$P(0|1) = 0$$

An erroneous threshold crossing is called an anamoloy (Ref 13, 15). For the given receiver, the probability of at least one anamolous pulse $P\{A\}$ in a frame time is equal to one. The M bit times are independent observation intervals, therefore, the probability of observing $r < M$ erroneous threshold crossings may be expressed in terms of the crossover probability, P_{FA}.

$$P(R \leq r) = \sum_{K=1}^{r} \binom{M-1}{K} P_{FA}^{\;K} \, (1-P_{FA})^{(M-1)-K} \tag{10}$$

$$R < M \quad P_{FA} \neq 1.0$$

Equation (10) merely demonstrates that given a particular P_{FA} as determined by the receiver threshold and the channel statistics, a figure of anamoly or measure of anamolous error is available.

To complete the discussion of the receiver structure, assumptions regarding bit times must be made. A complete summary of the receiver is also included in this paragraph. The assumptions regarding bit times pertain only to eliminating or minimizing any concern for demodulation errors other than those attributable to multiple anamolies. First, the quantization of frame times into bit intervals allows for sufficient guard time. This assumption minimizes the effect of intersymbol interference in that a pulse in frame KT will not carry over into frame (K+1)T. Figure 10 depicts this.

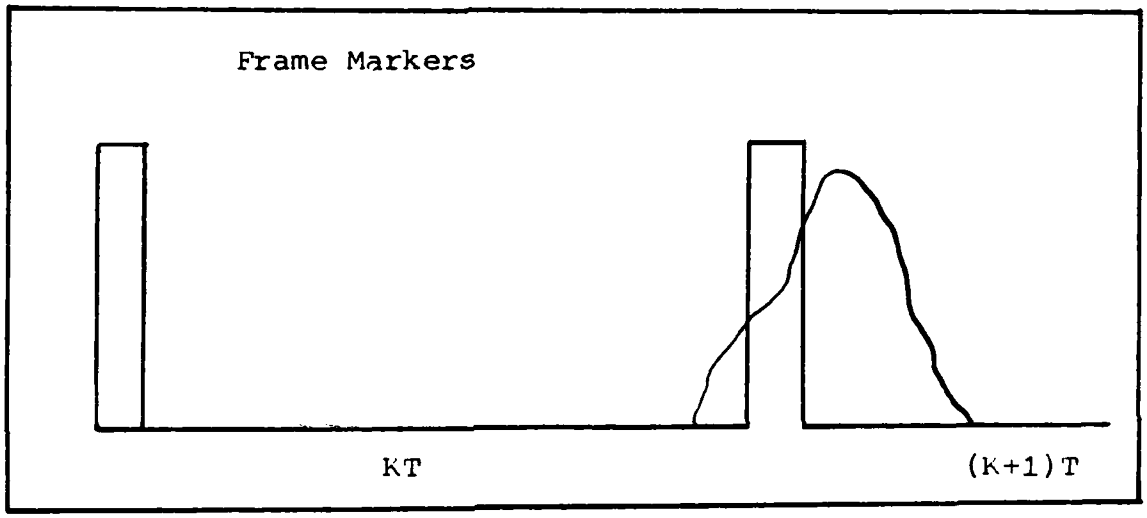


FIGURE 10. Intersymbol Pulse Interference

Secondly, during a single bit time, only one threshold crossing (pulse) may be observed. A pulse, noise or signal, is assumed to exist completely within a given bit time. This assumption eliminates any concern for any interbit interference.

The second assumption is fundamental to describing the receiver mode of operation. The receiver mode is best referred to as a "bit lock-out" mode of operation. That is, one and only one pulse may be detected per bit interval.

The first threshold crossing per bit interval constitutes the detection of a pulse within that observation interval. All other threshold crossings within that observation interval are ignored. The "bit lock-out" mode applies to each of the M observation intervals per frame.

Lastly, in addition to guaranteeing the existence of the actual data pulse within each frame, the original quantized pulse position is also preserved. A "1" transmitted in the ith bit position, will be detected in the ith bit position. This assumption is in addition to the probability of detection, $P_D = 1.0$ requirement. The receiver structure is best summarized as follows:

(1) The receiver makes M independent threshold observations per frame time.

(2) One threshold crossing per M observations must be the signal pulse with its position preserved over the channel.

(3) Interbit and intersymbol interference are negligible.

Before proceeding with the next section, the last term of (2) must be defined. It is convenient to rewrite (2) as:

$$\underline{Z} = \underline{S}_i + \underline{e}_j \qquad (11)$$

where $\underline{Z}$ and $\underline{S}_i$ are as previously defined and $\underline{e}_j$ is defined as an M-dimensional error vector due to channel noise. Each element of $\underline{e}_j$ is independent and, therefore, the set of error vectors, E, contains 2^M vectors:

$$E = \left\{e_1, e_2, \ldots \ldots e_\gamma\right\} \quad \gamma = 2^M \tag{12}$$

Equation (2) becomes:

$$\underline{Z} = \underline{S}_i + \underline{e}_j \qquad \begin{array}{l} i = 1, 2, \ldots M \\ j = 1, 2, \ldots 2^M \end{array} \tag{13}$$

The system model may be described as:

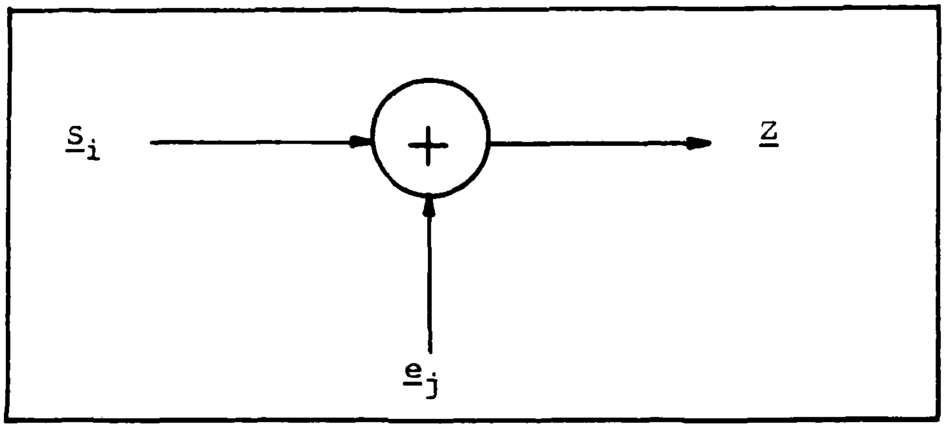


FIGURE 11. General Channel Model

The adder in Figure 11 is more accurately defined as an OR-ing device. The signal vector is OR-ed with the error vector. Given a signal vector $\underline{S}_i$, the observation vector $\underline{Z}$ may assume any one of 2^M-M+1 forms. The assumption that the pulse position or 1-bit position of the signal vector is preserved, establishes the fact that a given "1"-bit position cannot be changed, therefore, disallowing one signal vector being changed to another signal vector, hence the -(M-1) term.

On a frame-by-frume basis, Figure 9 expands to Figure 12 representing the M-hypothesis, 2^M-M+1 transitions, and 2^M possible observations.

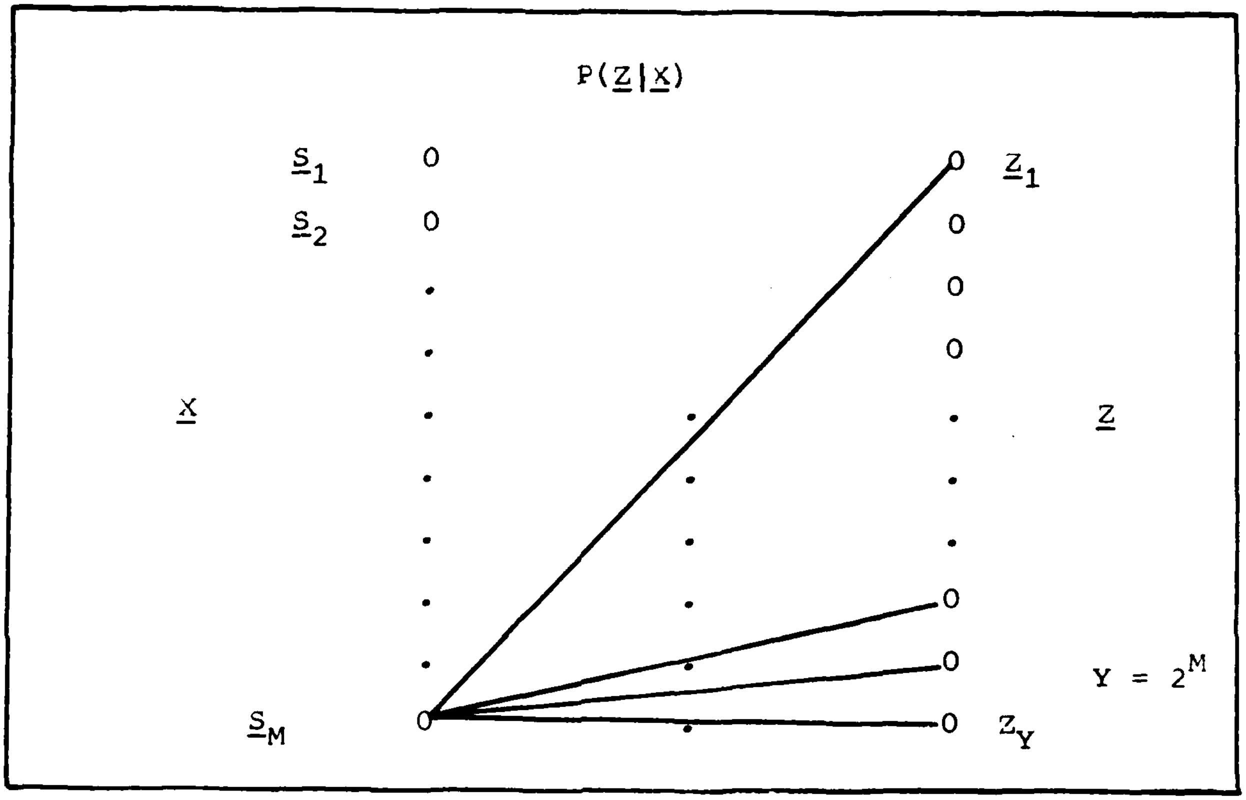


FIGURE 12. Discrete Vector Channel Model

By recalling independent bit times, the transition probabilities $P(\underline{Z}|\underline{X})$ may be expressed as:

$$P(\underline{Z}_K|\underline{X}_j) = \prod_{i=1}^{M} P(Z_i|X_i) \qquad (14)$$

where the $P(Z_i|X_i)$ are the bit crossover probabilities of Figure 9.

A receiver structure has been proposed in which the signal is assumed to be represented as a finite state (amplitude) sequence. This representation has been accomplished by quantizing the PPM frame times into M uniform subintervals or bit times. The signal pulse may occupy one and only one of these subintervals. The receiver maps the j^{th} bit time into the j^{th} discrete amplitude. The next section will discuss the Markov signal model in terms of the receiver structure.

SIGNAL MODEL:

The Viterbi decoding algorithm is indeed a maximum-likelihood decoder and, therefore, always optimum (Ref 2, 9). If the signal to be estimated (with respect to phase, frequency, amplitude, etc.) is modeled as a discrete time, finite state Markov process, in memoryless noise, then the Viterbi algorithm is a recursive optimal solution to the problem of estimating the state sequence of the discrete-time finite-state Markov process. Additionally, this solution may be viewed as a solution to the maximum a posteriori probability (MAP) estimation of the state sequence (Ref 2). The Viterbi algorithm metric (to be defined later) uses a form involving the state transition probabilities which characterize a discrete-time finite-state Markov process (Ref 10). This section uses the underlying Markov requirement as stated above, as a starting point for selection of a suitable signal model.

In order to develop a model of the amplitude sequence, it is beneficial to first limit the type and behavior of the message signal. The message is assumed to vary slowly in time. Frequencies to be expected range from less than 10 Hertz (Hz) to at most 100Hz. The signal is assumed to represent physical phenomenon that cannot change rapidly in time, nor can it experience an instantaneous change in performance values.

~~In the~~ previous section, a receiver structure was proposed to approximate the continuous amplitude sequence as a finite-state, amplitude sequence. The input sequence to the channel may be thought of as a finite-state-sequence of length N, of M possible states. The input sequence is assumed to start at time KT, K=1 and stop at KT, K=N where K is the K^{th} sample of the input sequence. The complete input sequence may be represented as:

$$U_1^N = (U_1^j, U_2^j, \ . \ . \ . \ U_N^j) \quad \forall \ j \in \left\{1, 2, 3...M\right\} \qquad (15)$$

Equation (15) describes the sequence of discrete pulse positions/amplitudes corresponding to some message m(t). To implement the Viterbi algorithm as a MAP estimator, the finite state, discrete time sequence must now be expressed as a first order Markov process.

In view of the now discrete nature of the system and the assumed behavior of the message m(t), a class of first order Markov processes known as birth-death processes, was selected for initial study. A general discussion of Markov processes follows:

(1) **Markov Process.** A process termed Markov is generally described in terms of a transition probability distribution on the states that compose the process (Ref 3, 4, 5, 6). A process is first order Markov in the sense that the probability:

$$P\left\{X^j_{K+1} \middle| X^i_0,\ X^i_1 \ldots\ldots X^i_K\right\} \tag{16}$$

of being in state X^j at time K+1 given all the states X^i up to and including time K, depends only on the state X^i at time K. Equation (16) becomes:

$$P\left\{X^j_{K+1} \middle| X^i_0,\ X^i_1 \ldots X^i_K\right\} = P\left\{X^j_{K+1} \middle| X^i_K\right\} \tag{17}$$

The term on the right side of the equality is the state transition probability for state X^i to X^j. The transition probabilities may be time varying, however, for this study, the transition probabilities are time invariant.

The transition probabilities for a Markov process may be expressed in the form of a stochastic matrix:

$$\bar{P} = (P_{ij}) \tag{18}$$

Each P_{ij} specifies the probability of making a transition from state i to state j, from one sample time to the next. For an M state process, the stochastic matrix $\bar{P}$ has dimension MxM such that:

$$\bar{P} = \begin{bmatrix} P_{ii} & \cdots & \cdots & P_{im} \\ \vdots & \ddots & & \vdots \\ \vdots & & P_{ij} & \vdots \\ \vdots & & \ddots & \vdots \\ P_{mi} & \cdots & \cdots & P_{mm} \end{bmatrix} \tag{19}$$

where:

$$o \leq P_{ij} \leq 1 \qquad 1 \leq i,\ j \leq M \tag{20}$$

and:

$$\sum_{i=1}^{M} P_{ij} = 1 \qquad J = 1,\ 2,\ \ldots .\ M \tag{21}$$

It is important to consider three qualities of the $\overline{P}$ matrix in relation to the finite state input sequence. First, the $\overline{P}$ matrix is time invariant and, therefore, the process is said to be stationary (Ref 6). That is:

$$P\left\{X(K+\gamma)=i \,\middle|\, X(N+\gamma)=j\right\} = P\left\{X(K)=i \,\middle|\, X(N)=j\right\} \quad N=K-1 \tag{22}$$

The process is said to be irreducible in that all states communicate. Communicate means that each state may be arrived at given enough time, independent of the starting state. If the $\overline{P}$ matrix was reducible, a given starting state for the signal process would rule out some subset of states. The $\overline{P}$ matrix would vary depending on the starting state. Sufficient information about a particular message sequence may yield a reducible stochastic matrix, however, for this study, no such prior information is available and, therefore, the assumption that all states communicate will hold. Lastly, all states are considered to be transient. No state is considered absorbing:

$$P\left\{X_i(K) \,\middle|\, X_j(K-1)\right\} > 0 \qquad \begin{array}{l} \forall\, i=1,2,\ldots\ M \\ \forall\, j=1,2,\ldots\ M \\ \forall\, K \leq N \end{array} \tag{23}$$

where N is the last state in the sequence.

The notion of transient and absorbing states is better understood in terms of a state transition diagram. In addition to a stochastic matrix, an M-state Markov process may be represented graphically by a state transition diagram similar to Figure 13.

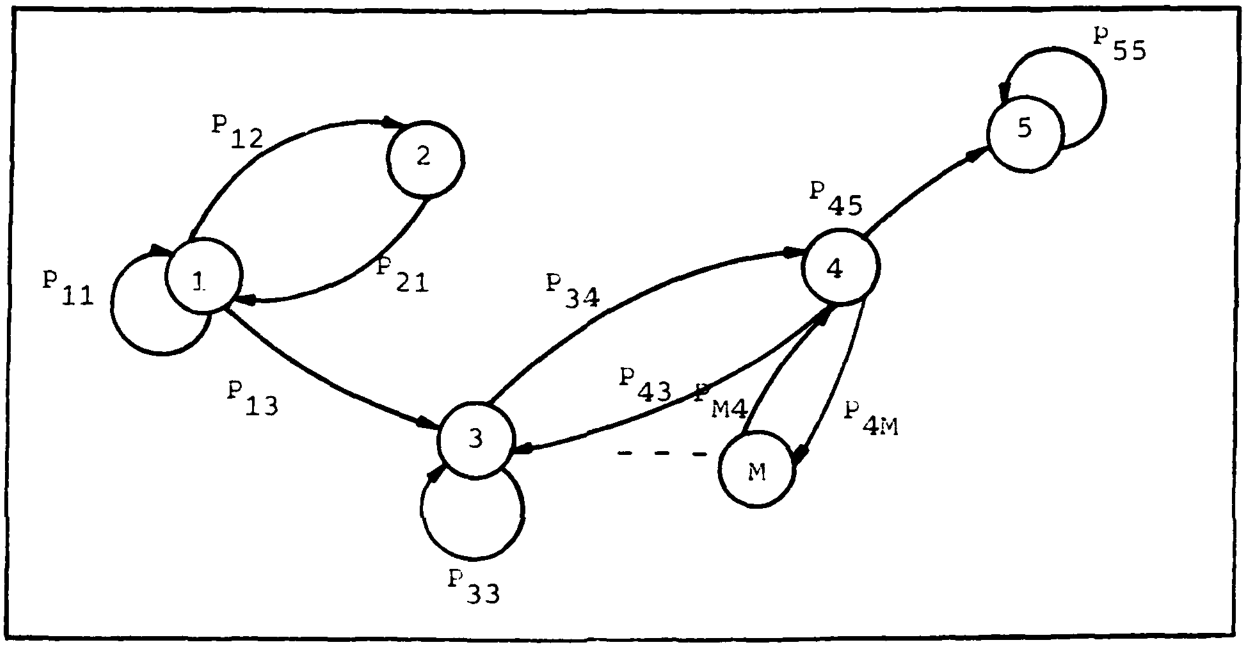


FIGURE 13. General Transition Diagram

The transition diagram resembles flow graphs used to describe a finite state machine, with the transition probabilities, P_{ij}, replacing the usual input/output information.

Figure 13 shows an example of both transient and absorbing states. States #1-#4 are transient in that, once any of these states are arrived at, there exists some probability of leaving that state. State #5 is an absorbing state in that once state #5 is arrived at:

$$P_{5j} = 0 \qquad \forall j \neq 5 \qquad (24)$$

and:

$$P_{55} = 1 \qquad (25)$$

The process stays in this state once it is reached. Examples of finite-state Markov processes that have both transient and absorbing states are finite-population pure birth processes and pure death processes. Other examples include the gambler's ruin chain, the absorbing state corresponds to the state in which the gambler has run out of money (Ref 3). The assumption that no absorbing state exists does not restrict the "behavior" of the process. A finite-state process with no absorbing states may indeed reside in the same state from sample to sample. However, the probability of making a transition out of this state remains greater than zero. In this manner, the Markov model used will not restrict the "behavior" of the message sequence. The model may be used to describe a message that is periodic or a message that experiences an absolute maximum or minimum. A Markov model with an absorbing state might be used to describe a message involving a known, steady state, constant value. A Markov model with only transient states will be used in an attempt to describe a more general message "behavior."

As will be demonstrated in the last section of this chapter, the $\overline{P}$ matrix is critical in implementing the Viterbi algorithm. The stationarity of the process requires only one stochastic matrix to be computed for given sequence. Parameters used to define a specific $\overline{P}$ matrix are described next.

The assumptions made regarding the transmitted message, m(t), makes the modelling of the sampled and quantized version of m(t) as a first order Markov process intuitively satisfying. Restricting the message behavior (i.e., rate of change with time, etc.), ensures that the message may be sampled fast enough to ensure that a sample value (quantized) at time KT is dependent only on the sample value (quantized) at time (K-1)T. Such a sample rate guarantees that a state sequence based on quantized amplitudes (discrete states) remains first order Markov in its characteristics. This sample rate is assumed to be achievable and will limit state (quantized amplitude) transitions for $(K-1)T \leq t \leq KT$ (i.e., from one sample time to the next) to one of the following possibilities:

(1) A transition may occur only from a given state to one of two adjoining states, or

(2) No transition occurs; the process remains in the given state.

Since each state corresponds to a quantized amplitude value, the quantized version of m(t) may change by at most one quantization level (or pulse position interval) from sample to sample. The sample rate must provide this limitation.

The state diagram of Figure 14 may be used to describe the Markov model and illustrate the adjoining state transition limitation.

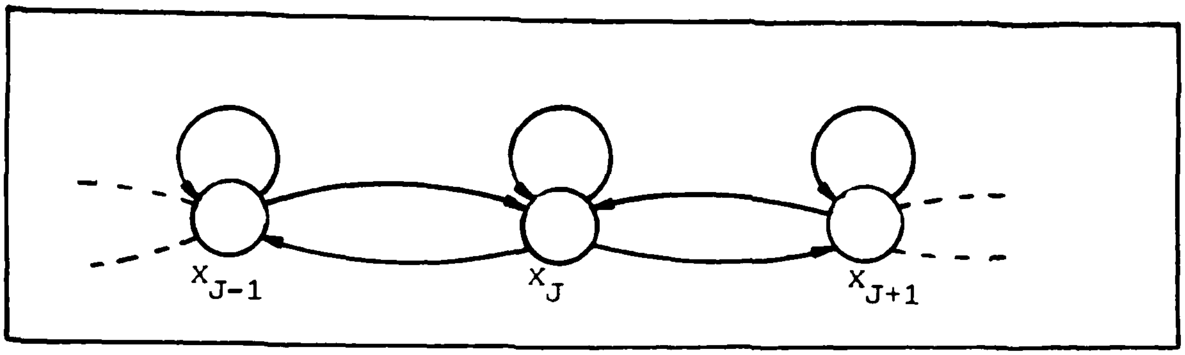


FIGURE 14. State Transition Diagram - Markov Process

The $\overline{P}$ matrix of equation (19) simplifies to a M x M matrix with nonzero elements only on the main and two adjoining diagonals:

$$\begin{bmatrix} P_{11} & P_{12} & 0 & \cdots & \cdots & 0 \\ P_{21} & P_{22} & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & P_{(M-1)M} \\ 0 & \cdots & \cdots & 0 & P_{M(M-1)} & P_{MM} \end{bmatrix} \tag{26}$$

where:

$$0 \leq P_{ij} \leq 1 \qquad |i-j| \leq 1 \tag{27}$$

$$P_{ij} = 0 \qquad |i-j| > 1$$

and:

$$\sum_{i=1}^{M} P_{ij} = 1 \qquad j=1,2,\ldots\ldots M \tag{28}$$

The restrictions and characteristics presented thus far reduce the signal model problem to selecting a first order

Markov process that fits the signal sequence as described by the state diagram of Figure 14, and the stochastic matrix $\overline{P}$. The process chosen is a special case of discrete time Markov process known as the birth-death process. Characteristics of the process are presented next.

(2) Birth-Death Process. A birth-death process (BDP) is a first order Markov process used to model changes in the size of a population or que. For a birth-death process, the system is said to be in state X_J when the population consists of J members. A key assumption in a BDP is that changes in population size occur by at most one person. A "birth" will change the population size to one greater and a "death" will change the size to one less. Multiple births or bulk disasters are not permitted. This is merely the adjoining state premise stated in equation (25).

A BDP is a homogeneous (stationary) process with the transition probabilities defined as (Ref 5):

$$P_{ij} = \begin{cases} d_i & j=i-1 \\ 1-b_i-d_i & j=i \\ b_i & j=i+1 \\ 0 & \text{elsewhere} \end{cases} \qquad (29)$$

The d_i term represents the probability that from one sample time to the next, a single death will occur, thus driving the population down to i-1 given the population is at i. Similarly, the b_i term represents the probability that single birth will occur, thereby increasing the population size up to i+1 given it is at size i at the previous sample time. The term $1-b_i-d_i$, is the probability that neither of these events occur from one sample time to the next. Clearly the conditions of equations (25) and (26) hold for a discrete time birth-death process.

Two special situations arise in a BDP. First, a death is not permitted with a population of zero. Additionally, contrary to intuition, a birth is allowed within a zero population level. This becomes more meaningful when the general BDP is refined further to one with specific transition probabilities (queing theory model). Specifically:

$$d_o = o \qquad b_o > 0 \tag{30}$$

Equation (29) will obviously produce the desired tri-diagonal stochastic matrix. The stochastic matrix of equation (24) becomes:

$$\overline{P} = \begin{bmatrix} 1-b_o & b_o & 0 \cdots\cdots & 0 \\ d_i & 1-b_1-d_1 & b_1 & \cdot \\ 0 & d_2 & 1-b_2-d_2 \quad \cdot & \cdot \\ \cdot & 0 & d_3 \quad \ddots & \cdot \\ \cdot & \cdot & 0 \quad \ddots & 0 \\ \cdot & & \ddots & \\ 0 & 0 & 0 \cdots\cdots & 4-d_M \end{bmatrix} \tag{31}$$

No births are allowed once the population has matured to its limit (finite population size), therefore:

$$P_{MM} = 1 - d_M \tag{32}$$

(3) M/M/K//M Model. Numerous derivatives of the general class of birth-death processes exist. Many adaptations to the BDP are utilized in queing theory. One such adaptation was chosen for the manner in which it meets the signal sequence assumptions. The model, in queing theory terminology, is an M/M/K//M queing model (Ref 5). This model describes a system of finite customer population and a finite number of servers. For this study, the number of servers is limited to the same size as the population.

The finite population has as its limit M customers, corresponding to M possible pulse positions per frame. The population size of the que defines the state of the process. Each customer has an "arriving" parameter λ. The system also has M servers available each with a "serving" parameter μ. These parameters, λ and μ, represent birth and death rates per unit time with respect to a general BDP. Before proceeding, it is necessary to introduce what is called the set of birth death coefficients.

Regarding the nature of births and death, the notion of a birth rate λ_K describes the rate at which births occur when the population is of size K. Similarly the notion of a death rate μ_K describes the rate at which deaths occur when the population is of size K. These rates do not change with time; however, they are a function of the current state X_K (current population size). Referring back to the general birth-death process for a moment, the transition probabilities of equation (27) may be expressed in terms of the birth and death coefficients. The probabilities become:

$$b_K = P_r \left\{ \text{exactly 1 birth in T sec} \middle| \text{K in pop} \right\} \qquad (33)$$
$$= \lambda_K \; T$$

$$d_K = P_r \left\{ \text{exactly 1 death in T sec} \middle| \text{K in pop} \right\}$$
$$= \mu_K \; T$$

The time T, is the step time or sample time of the population (que). A sample occurs every T seconds. The sample time T is assumed small enough to ensure that the limits of a transition are ± 1 in population size. This, in turn, satisfies the nearest neighbor or adjoining state limitation stated earlier with respect to the message sequence.

The sample interval T, is a critical factor in the application of this model. A sample rate, $f_S = 1/T$, that produces a change of at most one pulse position (from sample to sample) is the foundation upon which selection of a particular Markov model rests. The "fast" sample rate allows the sequence to be modelled a sequence of adjoining, discrete amplitude/pulse-position states. The benefits of the adjoining state requirement will be graphically illustrated in the next section where the concept of a trellis diagram is introduced.

The birth-death coefficients that describe the finite customer population, finite server model are as follows:

$$\lambda_K = \begin{cases} \lambda(M-K) & 0 \leq K \leq M \\ 0 & \text{otherwise} \end{cases} \qquad (34)$$

$$\mu_K = K\mu \qquad K=1,2,\ldots\ldots M$$

This system is homogeneous and irreducible. Additionally, all states are transient and no state is absorbing (Ref 5). The transition probabilities and hence the behavior of the process is completely specified in terms of:

(1) The arrival or birth rate, λ (35)

(2) The serving or death rate, μ

(3) The sample rate, 1/T

(4) The size of the population, $K \leq M$

These four parameters will be used to specify the signal sequence. The finite population corresponds to the finite number of pulse positions, the arrival and serving rates represent rates of signal change per unit time, and $f_S=1/T$ is the sampling rate. For a given sample rate 1/T and M pulse positions, the rate parameters will be varied over a finite range and the adequacy of the model will be observed in the implementation of the Viterbi algorithm. The next section deals with the Viterbi algorithm.

VITERBI ALGORITHM:

In the previous section, the importance of specifying the process in terms of a set of transition probabilities was reserved for this section. This section will establish the necessary terms and the importance of the transition probabilities in implementing a MAP estimator by the Viterbi algorithm.

Hence forth, the state diagram of Figure 14 will no longer be used to describe the amplitude state sequence of the signal. Instead, a new convention, the trellis diagram, will be introduced in this section. The convention of a trellis diagram is extremely useful when using the Viterbi algorithm.

(1) Trellis. In terms of the discrete time BDP presented in the previous section, a trellis diagram expresses the possible behavior of the process for a given starting state. (Also, see Appendix D for discussion of expanded trellis).

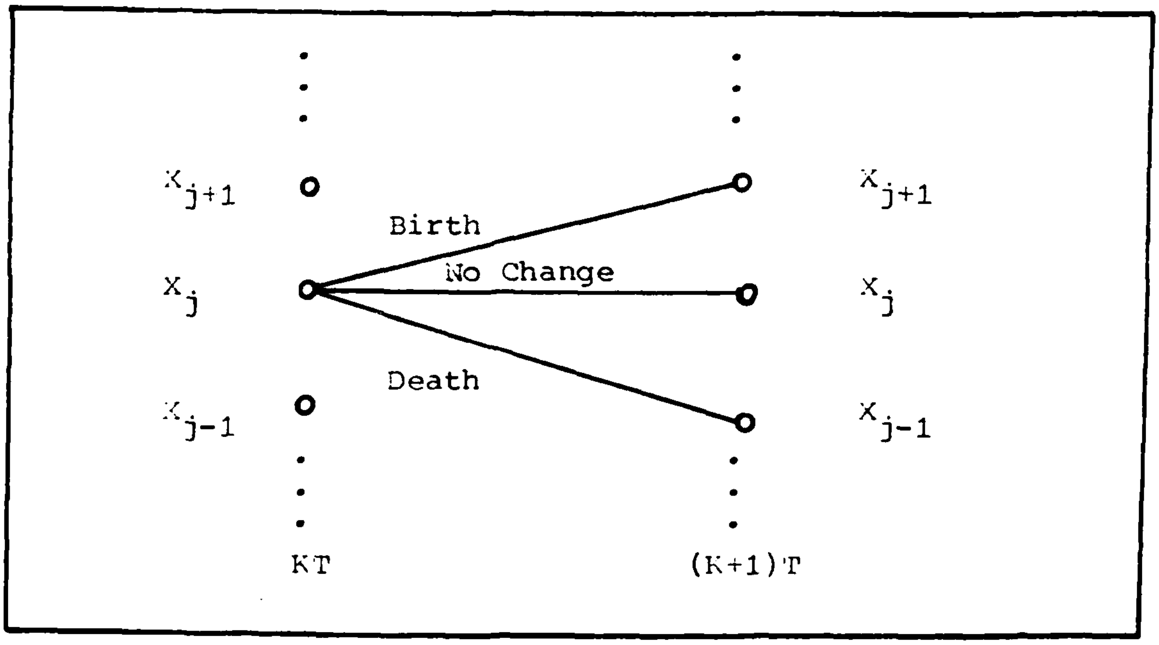


FIGURE 15. State Representation

Figure 15 depicts a portion of a trellis for two sample times and three of the M states. The paths connecting the states represent the possible transitions that the process may undergo from one sample to the next. In terms of the signal amplitude sequence, X_j represents one of M amplitude states and KT represents the K^{th} sample time.

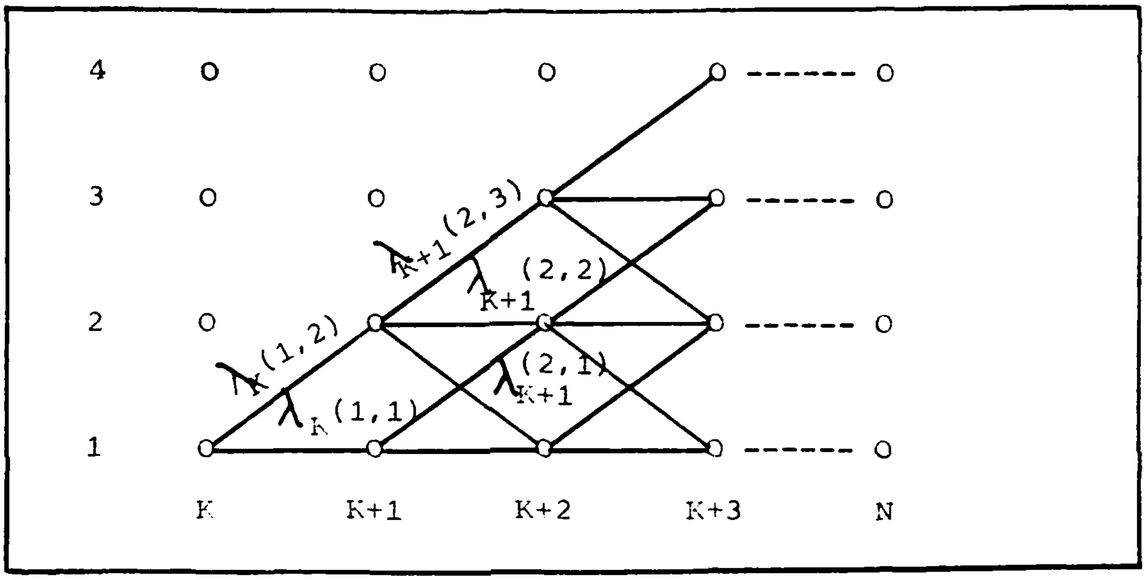


FIGURE 16. Trellis Representation of State Transitions

Figure 16 shows a trellis representing a system of four states over four sample times. Given the starting state shown, the sequence will follow one and only one path through the trellis based on the definitions and assumptions presented in the previous section. The correct path is described by equation (15), the input amplitude sequence. Given the transition probabilities of the sequence and the observation vector $\underline{Z}$, the Viterbi algorithm, in providing the MAP estimate of the amplitude sequence, will in reality, be finding the shortest path through the trellis. To find the shortest path, the algorithm requires a metric to be assigned to each branch.

Branch metrics are shown in Figure 16. A branch metric:

$$\lambda_K (j, 1) \tag{36}$$

is the metric or distance from state j to state 1 given the system is in state j at time K. For convenience, since only integer multiples of T are considered, each sample time will be expressed in terms of only an integer. A path through the trellis is described in terms of its length by the sum of its branches:

$$\Gamma_1^N \sum_{K=1}^{N} j_K(j, 1) \qquad \begin{matrix} \forall j = 1,2,....M \\ \forall 1 = 1,2,....M \end{matrix} \tag{37}$$

It is now convenient to define the branch metrics.

(2) <u>Branch Metric</u>. The desired result is an estimate of the Kth input sequence, $\left\{X_K\right\}_1^N$. This estimate is denoted as $\left\{\hat{X}_K\right\}_1^N$. It will be shown that the MAP estimate of the amplitude sequence is the sequence $\left\{\hat{X}_K\right\}_1^N$, that provides the shortest path through the trellis.

Forney (Ref 2) has shown a one-to-one correspondence to exist between a state sequence:

$$\left\{X_K\right\}_1^N = \left\{X_1, X_2 ....X_N\right\} \tag{38}$$

and the transition sequence:

$$\xi = \left\{\xi_1, \xi_2, \xi_3 \ldots\ldots \xi_N\right\} \tag{39}$$

The transition ξ_K is defined as:

$$\xi_K = (X_{K+1}, X_K) \tag{40}$$

Since the process is assumed to be observed in memoryless noise, there is a sequence $\left\{z_K\right\}_1^N$, of observations, in which z_K depends probabilistically only on the transition ξ_K at time K, or that:

$$P\left\{\left\{z_j\right\}_1^N \middle| \left\{x_j\right\}_1^N\right\} = P\left\{\left\{z_j\right\}_1^N \middle| \left\{\xi_j\right\}_1^N\right\} \tag{41}$$

$$= \prod_{K=1}^{N} P\left\{z_K \middle| \xi_K\right\}$$

A special case of this formulation is one in which the output of the memoryless channel z_K, depends only on the state x_K, and:

$$P\left\{\left\{z_j\right\}_1^N \middle| \left\{x_j\right\}_1^N\right\} = \prod_{K=1}^{N} P\left\{z_K \middle| x_K\right\} \tag{42}$$

The Viterbi algorithm becomes a solution to the following problem. Given a sequence of N observations $\left\{z_K\right\}_1^N$ of a discrete-time, finite state Markov process in memoryless (white) noise, find the state sequence $\left\{\hat{x}_K\right\}_1^N$ for which the "a posteriori" probability $P\left\{\left\{x_K\right\}_1^N \middle| \left\{z_K\right\}_1^N\right\}$ is maximum. This is merely the MAP decision rule. This rule is known to minimize the error probability in detecting the whole sequence and is, therefore, optimum in this sense (Ref 2).

The problem of maximizing $P\left\{\left\{X_K\right\}_1^N \middle| \left\{Z_K\right\}_1^N\right\}$ is equivalent to finding the sequence for which:

$$P\left\{\left\{X_K\right\}_1^N, \left\{Z_K\right\}_1^N\right\} = P\left\{\left\{X_K\right\}_1^N \middle| \left\{Z_K\right\}_1^N\right\} P\left\{\left\{Z_K\right\}_1^N\right\} \qquad (43)$$

is maximum. This, in turn, is equivalent to minimizing:

$$-\ln P\left\{\left\{X_K\right\}_1^N, \left\{Z_K\right\}_1^N\right\} \qquad (44)$$

Since equation (44) is a monotonic function and to every possible state sequence $\left\{X_K\right\}_1^N$ there corresponds a unique path Γ_K through the trellis, there is a one-to-one correspondence between paths and sequences.

From Bayes Rule, equation (43) becomes:

$$P\left\{\left\{X_K\right\}_1^N, \left\{Z_K\right\}_1^N\right\} = P\left\{\left\{X_K\right\}_1^N\right\} P\left\{\left\{Z_K\right\}_1^N \middle| \left\{X_K\right\}_1^N\right\} \qquad (45)$$

By invoking the Markov and memoryless properties, equation (45) becomes:

$$\prod_{j=1}^{N} P\left\{X_{j+1} \middle| X_j\right\} \prod_{j=1}^{N} P\left\{Z_j \middle| X_{j+1}, X_j\right\} \qquad (46)$$

Equation (44) is therefore:

$$\sum_{j=1}^{N} \left[-\ln P(X_{j+1}|X_j) - \ln P\left\{Z_j \middle| \xi_j\right\}\right\} \qquad (47)$$

Hence, each branch metric is assigned the length:

$$\gamma(X_j) = -\ln P\left\{X_{j+1} \middle| X_j\right\} - \ln P\left\{Z_j \middle| X_j\right\} \qquad (48)$$

and the total path length corresponding to the sequence $\left\{X_K\right\}_1^N$ is:

$$-\ln P\left\{\left\{X_K\right\}_1^N, \left\{Z_K\right\}_1^N\right\} = \sum_{j=1}^{N} \gamma(X_j) = \Gamma_K \qquad (49)$$

The relation of equation (49) points out that Γ_K may be computed recursively. Given the branch length of equation (36), the length of path K, at time j is:

$$\Gamma_K\,(j) = \Gamma_K\,(j-1) + \gamma_j(m,\ 1) \qquad (50)$$

The expressions in equations (46) and (47) completely define the branch metrics. The left hand term represents the transition probabilities of the Markov signal process. The importance of specifying the process transition probabilities is now evident. To compute the MAP estimate, the extent of prior knowledge required concerning the signal process is simply the transition probability values expressed in the $\overline{P}$ matrix.

The right hand term of equations (46) and (48) is derived from the statistics of the channel. The $P\left\{Z_K \middle| X_K\right\}$ depends upon the received data. For a Guassian channel, $-\ln P\left\{Z_K \middle| X_K\right\}$ is proportional to $(Z_K-X_j)^2$. Using the vector representation presented earlier, $-\ln P\left\{Z_K \middle| X_K\right\}$ is proportional to:

$$\left\| \underline{Z}_K - \underline{X}_j \right\|^2 \qquad \forall\ j=1,2,.....M \qquad (51)$$

where:

$$\left\| \underline{X} \right\|^2 = \sum_{i=1}^{M} X_i^2 \qquad (52)$$

(See Appendix A for a similar discussion, using a Poisson channel). Equation (51) represents a comparison of the received vector with each of the possible signal vectors. The branch metric at time j becomes:

$$\gamma_j(m,1) = -\ln P\left\{\underline{X}_j \succ \underline{1} \,\middle|\, \underline{X}_{j-1} = \underline{M}\right\} + \left\|\underline{Z}_j - \underline{X}_M\right\|^2 \qquad (53)$$

Let the estimate at time j be represented by $\hat{X}(j)$ in order to examine the recursive nature of the trellis path length computation. The sequence $\left\{X_K\right\}_1^N$ which minimizes Γ_K while passing through $\hat{X}(j-1)$ on its way to $\hat{X}(j)$ must do so by following sequence $\left\{X_K\right\}_1^{j-2}$ which minimizes $\Gamma_K(j-2)$. If it did not, a better sequence $\left\{\hat{X}_n\right\}_1^{j-2}$ could be chosen.

To extend the MAP estimate to time j+1, the estimator must take the new measurement $\underline{Z}(j+1)$ and, for each amplitude state $X(j+1)=\xi_M$, $M=1,2,\ldots\ldots M$, calculate the $\Gamma_K(j+1)$ for each sequence $\left\{\hat{X}\right\}_1^j$ plus $X(j+1)=\xi_M$. For each value of M, the minimum path length, $\Gamma_K(j+1)$ is selected and the value of $\hat{X}(j+1)=\xi_M$ is concatenated with the sequence $\left\{\hat{X}\right\}_1^j$ which ends in $X(j)=\xi_1$. The sequence $\left\{\hat{X}_K\right\}_1^{j+1}$ has then been determined. At the end of the calculation, the observation $\underline{Z}(j+1)$ is no longer required and, therefore, discarded. There will be M amplitude sequences stored, one ending in each of the M possible discrete amplitude states. The M sequences may be extended another step in time by taking the measurement, $\underline{Z}(j+2)$, and repeating the calculations. Therefore, at time jT (or simply j), it is only necessary to store these M $\left\{\hat{X}_K\right\}_1^j$, $K=1,2,\ldots\ldots M$; the M values of $\Gamma_K(j)$ corresponding to the M sequences; the current time counter j; and the transition probability values (Ref 14).

The minimization may be stated as follows:

$$\underset{\{\hat{X}\}_{N-1}^{N}}{\text{Min}} \left[\underset{\{\hat{X}\}_{1}^{N-2}}{\text{Min}} \left[\Gamma(j-1) + \gamma_j \ (M,1) \right] \right] \qquad (54)$$

or in algorithm form:

```
For each time j = 1, ..... N
Begin
    Obtain the current observation $\underline{Z}_j$
    For each amplitude state $\xi_M$, M=1,2,....M
    Begin
        For each amplitude state $\xi_1$, 1=1,2,....N
        Begin
            Calculate $\Gamma_M(j)=\Gamma(j-1)+\gamma_j(m,1)$
        End
        Pick the smallest $\Gamma_M(j)$
        Concatenate $\hat{X}(j)=\xi_M$ with the sequence
          ending in $\hat{X}(j-1)=\xi_1$
        Set $\Gamma(j)=\Gamma_M(j)$ for the sequence $\{\hat{X}\}_1^j$
          ending in $\xi_M$
    End
End
```

At each time j, there will be a sequence of states ending in each of the M possible states. These are called survivor sequences. In general, there will be survivor sequences from the previous time (j-1) which are not concatonated with state value to form a current survivor sequence. These unused sequences need not be retained.

(2) Decision Depth. The Viterbi algorithm used for other than the usual convolutional coding application becomes a problem of dynamic program. In particular, the algorithm is a backward dynamic programming method. The term backward applies to the manner in which estimations are arrived at.

The state values stored in the M sequences tend to converge to a single state if the sequences are inspected long enough in time. Because of this convergence, at time j, it is possible to determine the MAP estimate of the amplitude sequence $\hat{X}(j-J_o)$, by selecting the sequence $\left\{\hat{X}_K\right\}_1^j$ with the smallest Γ_K and outputting the $\hat{X}(j-J_o)$ value from that sequence. If all the M sequences have not converged to a single state $\xi_M=\hat{X}(j-J_o)$, then J_o may be made larger. Making J_o larger only guarantees that $\hat{X}(j-J_o)$ is the MAP estimate of the true sequence value, not that $\hat{X}(j-J_o)$ is the closest neighbor to $X(j-J_o)$.

The number J_o is the fixed lag of the system and is referred to as the decision depth. Simply stated, the decision depth is the time lag before an estimate is output. Because the estimate is made back in time from the current observation as determined by the decision depth, the term backward dynamic programming applies.

The complexity of the estimator may be seen from the algorithm statement. The majority of computing time is spent in calculation of the values of $\Gamma_K(j)$. These values must be calculated M times for each of the M possible states, at each time jT j=1,2,.....N. The complexity in terms of computation time is proportional to NM^2. As previously stated, at time j, the amplitude may take on one of M values. A sequence composed of N values leads to M^N possible sequences to search through in order to find the one that minimizes Γ_K (the K^{th} path) equation (49). As M and N become large, brute force, global searches become impractical. The recursive nature of equation (49) leads to the NM^2 term stated above. Additionally, the estimator outputs a sequence value with a fixed lag in time, as opposed to a global search method which must store an entire observation sequence and then determine the MAP estimate for the given observation sequence. For a finite time lag:

$$J_o << N \tag{55}$$

the storage requirements are also greatly reduced.

A receiver structure and signal model have been presented to describe a system involving a slowly varying message signal that has been pulse position modulated. The amplitude of the message may be represented as a sequence of quantized amplitude (pulse position) states. Using backward dynamic programming, it is possible to compute a maximum a posteriori estimate of the amplitude sequence. The question of how well the system performs is addressed in Chapter III.

III. ESTIMATOR SIMULATION:

A Monte Carlo simulation of the backward dynamic programming algorithm amplitude sequence estimator is useful to obtain performance data for an entire range of input data and estimator parameters. This chapter presents the simulation phase of this study. This chapter is organized into four sections: simulation setup (software considerations); a discussion of the assumed set of message signals; parameter variation; and simulation results.

SIMULATION SETUP:

The simulation program was written in FORTRAN V and run on the Data General Eclipse/250 system available for use in the Digital Signal Processing Laboratory at the Air Force Institute of Technology. FORTRAN V is a Data General version of FORTRAN. This version provides a great improvement in execution time over FORTRAN IV when used on the Eclipse.

(1) Software. The simulation program is divided into four main sections. Once the simulation parameters have been input, the program first computes the transition probability matrix. Next, the quantized version of the message is represented as a sequence of signal vectors as in equations (6), (7), and (8). Next, a noise vector is generated, and then added (OR-ed with) to the signal vector to produce the observation vector. The true pulse position/

amplitude sequence is represented as a sequence of N integers which correspond to one of M possible pulse positions for each of N samples. The pulse position sequence is then estimated over an observation period consisting of the N samples. The error and mean-square error are calculated for each sample. The estimation is repeated for many variations of a single message in order to calculate ensemble averages of error and mean-square error at each time interval (sample).

(2) Signal Set. A variety of message signals were constructed and used to test the estimator. Each message is derived from:

$$M_i(KT) = f(rKT) \tag{56}$$

where:

$$r = e^{-u} \qquad u \in (0, 1) \tag{57}$$

The term r, represents the "rate" of the signal $r \in (.368, 1)$. Using the exponential relationship precludes the possibility of the message having a rate close to, or exactly zero. The term KT represents the K^{th} sample time.

A collection of ten signals make up the signal set. These signals appear in Appendix B. Each signal has a magnitude less than or equal to one and this range has been quantized into M=32 amplitude states. The shape of each signal is determined by the rate parameter, r. By virtue of the generation method, the rate parameter also determines the starting state. The signals are shown for extreme values of r. Each pair of figures shows graphically the affect the rate parameter has on the overall shape of the signal as well as its affect on the starting state. The random starting state reflects more accurately, a realistic signal and will demonstrate estimator effectiveness for an unknown starting state within the trellis. Each signal has a peak amplitude of 1.0.

The relationship used to generate each signal is also provided in Appendix B. It is worthy of note that only signal #3, the sinewave has a random shift. This random shift provides the random starting state desired for each simulation. The assumed starting state used for simulations involving signal #3 was ξ_1=16.

Lastly, for signals #6-#10, a fixed time delay of five seconds (one half the observation period) was used. The maximum occurs at this time for each signal (#6-#10) and, therefore, the ensemble average state error and ensemble average mean-square error graphs will reflect the estimator performance during this known transition period. The ability of the estimator to adjust to a transition (change of slope) is an important aspect of the estimator's performance and, therefore, only a fixed time delay was used so that the error performance was easily distinguishable from the startup (initial transient) error.

The estimate, as derived by dynamic programming (VA), is independent of the assumed starting state if the observation period is long compared to the number of possible states ($N \gg M$) (Ref 2). The algorithm must start without exact knowledge of the starting state. The algorithm is, therefore, initialized with a branch length $\Gamma(0)=0$ and an assumed starting state. After an initial transient, the startup error, there is a high probability that all survivors will merge with the correct path (Ref 2). The algorithm is said to be self-synchronizing.

The initial branch length, as well as the chosen initial state, may reflect any a prior knowledge of the message. The initial state ($\xi_1=5$) chosen for signals #6-#10 and ($\xi_1=16$) for signal #3 reflects an appropriate estimate of the initial state based on known signal performance. The state ($\xi_1=5$) reflects some a priori knowledge the signal may start close to, but not exactly zero amplitude. A similar argument holds for signal #3.

PARAMETER VARIATION:

Five parameters must be determined prior to each simulation; four (λ, μ, M, T), pertain to the signal model and one, (J_o) to the algorithm. The first four are used in calculating the transition probabilities (priors) used by the algorithm and must be considered in estimator performance. The fifth pertains to the memory established by the priors within the trellis and its branches. The term, J_o, determines the manner in which the algorithm uses this memory.

A discussion of the relationship between these parameters is presented in this section. This discussion provides the basis for the variations made in the parameters used for each simulation.

(1) Transition Probability Values. The transition probability matrix for M=32 state process becomes a 32 x 32 square matrix. The stochastic matrix is generated using equations (31) and (33). For signals #3 and #6-#10, the rate parameters of equation (33) are:

$$\lambda = \mu = r \qquad (58)$$

For signals #1 and #5, the rate parameters are:

$$\lambda = r \qquad (59)$$
$$\mu = .0001\ (\lambda)$$

and for #2 and #4:

$$\mu = r \qquad (60)$$
$$\lambda = .0001\ (\mu)$$

These parameters were based on general signal behavior and an attempt to describe the relative rate of change of the signal in terms of the queing model parameters.

Signals #1 and #5 may be described by a limiting case birth-death process. This limiting case is a finite state pure birth process. The rate parameters for a pure birth process involve only the arrival rate λ (μ= 0). The system expands from some population size up to its maximum (the maximum being an absorbing state). The value for the arrival parameter expressed in equation (59) is valid for the range of values of λ used and allows the natural log (ln) of the elements of the transition probability matrix equation (48) to be finite.

Similarly, signals #2 and #4 may be described by a pure death process. The only rate parameter needed is the serving rate, μ. This type of system starts at some population level and will decrease in size with each sample time. For this model, the relation of equation (60) is valid for the range placed on the values of r.

The remaining signals are more adequately described by equation (58). Neither μ nor λ are assumed to be zero. Additionally, the symetric nature of these remaining signals lead to the description of equation (58) with respect to the finite population queing model.

The parameter T, in equation (33) is the sample period. Expressions (33) and (34) imply that the number of states M and the rate parameters λ and μ, will determine the maximum value for T. Any relationship between λ, μ, M, and T must provide for two limiting conditions. First, the requirements of equations (27) and (28) must be satisfied. Secondly, the relationship must be such that the sample rate, $f_S=1/T$, is fast enough to provide some degree of memory for the algorithm in addition to limiting the change in value of the sampled signal to ± 1 amplitude states/pulse positions from sample to sample.

The sample rate provides the "memory" for the MAP estimator. The term memory is somewhat nebulous; however, it describes appropriately, the amount of information provided by the transition probability matrix of equation (31). An analogy may be derived from a discussion of bandwidth and Nyquist rate sampling.

A Nyquist sampled (band limited) signal, may be reconstructed by a low pass filter, if the cutoff frequency of the low pass filter is equal to the maximum frequency of the signal, f_S. A sample rate of less than $2f_S$ or a filter cutoff frequency not equal to f_S will produce aliasing.

The concept of signal reproduction via Nyquist sampling is not a concern, however, a relationship to indicate a lower bound on the sample rate that guarantees equations (27) and (28) and provides enough memory, is desired. The relationship hypothesized, in terms of the queing model parameter is given as:

$$K\ (\lambda_{max}\ M) \leq f_S \qquad K \geq 1 \tag{61}$$

Clearly equation (61) provides values of $f_S = 1/T$ that will satisfy equations (27) and (28) for $K \geq 1$. The question to be addressed is will equation (61) provide the needed memory, and over what ranges of receiver performance? It will be shown that equation (61) is adequate for specific receiver characteristics. Specifically, as the number of anamolous pulses, A, increases, the minimum sample rate of equation (61) is not adequate for all signal rates. A faster sample rate is required to increase the amount of memory. The value of K must be increased as the memory requirement increases.

The sample rate chosen initially was (K=1.5625):

$$f_S = 50 \text{ samples/sec (S/S)} \qquad (62)$$

Based on equation (61), this rate is adequate for:

$$\lambda \leq .78125 \qquad (63)$$

for some value of A. The sample rate of equation (62) was shown to be adequate for all $\lambda \in (.36, 1.0)$ if the number of anamolous pulses was less than 12. The inadequacy of equation (62) showed up first for signals with $\lambda > .78125$. The sample rate was then increased to 64 samples per second.

As the number of anamolous pulses is increased to A=22, the sample rate of 64 S/S (J_o=5) is adequate to provide a reasonable reconstruction for signals with $\lambda \leq .4$. That is, the value of K must be increased to 5, and from equation (61):

$$f_S \geq 160 \qquad (64)$$

There is obviously some limit to this relationship. As A increases, the algorithm must rely more on the priors to distinguish the shortest path from the entire set of paths. Each path length will vary by the contribution of the transition probability values (priors). Once A=31, the priors will not provide the distinction needed, no matter what the sample rate. Clearly, there must exist an upper limit on the sample rate. This upper limit will provide the memory for some upper limit on the number of anamolies, A. Determination of these upper bounds is beyond the scope of this study as is any expression relating the two values. An attempt is made, however, to find the value of A, at which a reliable reconstruction may no longer be achieved for a given ensemble of message signals and a given set of model and estimator parameters.

In the previous chapter, the idea of a decision depth was introduced. The decision depth and the assumed initial state are two algorithm parameters that may be adjusted independent of all other parameters.

In the context of a convolutional encoder/decoder, the memory, as provided by the encoder, is represented in the form of a constraint length (Ref 2). The decision depth is correspondingly, set at three to five times this constraint length. For this study, no such constraint length exists, as the memory is supplied by the sample rate. Instead, the notion of trellis maturation is used to describe a minimum as well as a sufficient decision depth.

Appendix D depicts a trellis diagram for a seven state process. The state transitions depicted by the branches follow the assumptions pertaining to the signal model. It is evident from Figure 174 that a "legal path" connecting all possible states does not exist until time (K+6)T. The trellis shows that the time required to provide a minimum of one "legal path" (see Appendix D) connecting all legal states is dependent upon the assumed initial starting state. This time will be referred to as the maturation time and will provide what will be hypothesized as the minimum decision depth. The minimum decision depth becomes:

$$J_o = M - \xi_1 \qquad 1 \le \xi_1 \le 16 \qquad (65)$$

and:

$$J_o = \xi_1 \qquad 17 \le \xi_1 \le 31 \qquad (66)$$

The range on J_o becomes:

$$16 \le J_o \le 31 \qquad \forall \; \xi_1 \qquad (67)$$

where again ξ_1 is the assumed starting state. It will be shown that the performance of the algorithm improves as the value of J_o is increased to the level specified by equations (65) and (66) for given values of A. For a given value of A, it will be demonstrated that making J_o arbitrarily large does nothing to improve performance. Specifically as A increases, increasing the decision depth fails to gain an improvement in algorithm performance.

(2) Sensitivity. In view of the previous discussions concerning model qualification, signal rate, and the manner in which they contribute to the prior knowledge used by the estimator, two facets of sensitivity will be addressed.

First, the sensitivity of the algorithm to the model used to compute the priors will be examined. In particular, what will happen if a signal model describing signals #1 and #5 is used to form the priors when the signal to be estimated is one of the other eight? Similarly, how will the performance of the estimator be affected if a pure death process is used to form the priors when the other types of signals are to be estimated? The variance in performance is referred to as model sensitivity.

Secondly, the sensitivity of the algorithm when a random rate parameter is used to derive the priors, will be demonstrated. Specifically, how will performance vary when the correct model is used; however, the rate parameter used to generate the signal is not equal to the rate parameter used to generate the transition probability matrix. Sensitivity of this type is referred to as rate sensitivity. Rate sensitivity is related to the discussions on the sensitivity of the Viterbi algorithm to source statistics (Ref 16) and, therefore, only a slight change in performance is expected.

The Monte Carlo simulation study was used to provide measures describing the average performance of the estimator. Additionally, the performance of the estimator may be examined for specific signals within the ensemble. Computer resource availability precludes an exhaustive study of parameter variation. The sample rate and decision depth must be considered the most critical elements toward providing a reliable MAP estimate. Therefore, the simulation results to follow will demonstrate an acceptable level of performance for specific values of sample rate and decision depth.

SIMULATION RESULTS:

The results of the estimator simulations are presented in this section and shown in the figures that follow. The parameter values and conditions are shown in the figures. The notion of an expanded trellis used for the computer simulations is discussed in Appendix D.

(1) Confidence of Sample Statistics. The data presented in the following figures are a result of taking ensemble averages of the mean-square error, as well as ensemble averages of the state error, over 25 Monte Carlo runs. This value was selected by observing how the average of the ensemble mean-square error, $\langle \overline{MSE} \rangle$ changes with increased number of runs, shown in Figure 17. As the number of runs increases, the ensemble average mean-square error tends to settle down to a specific value. The number 25 was selected to provide meaningful results at a reasonable expenditure of computer time. This number was reduced to 20 without any loss in accuracy when the sample rate was increased from 50 samples/second to 64 samples/second. Similar conclusions were drawn by observing the ensemble average of the state errors as the number of runs is increased.

The performance of the estimator is related to the ensemble average of the rate parameters used to generate the signals. As expected, the estimator's average performance is better when the average signal rate is low and the ensemble average performance worsens as the average signal

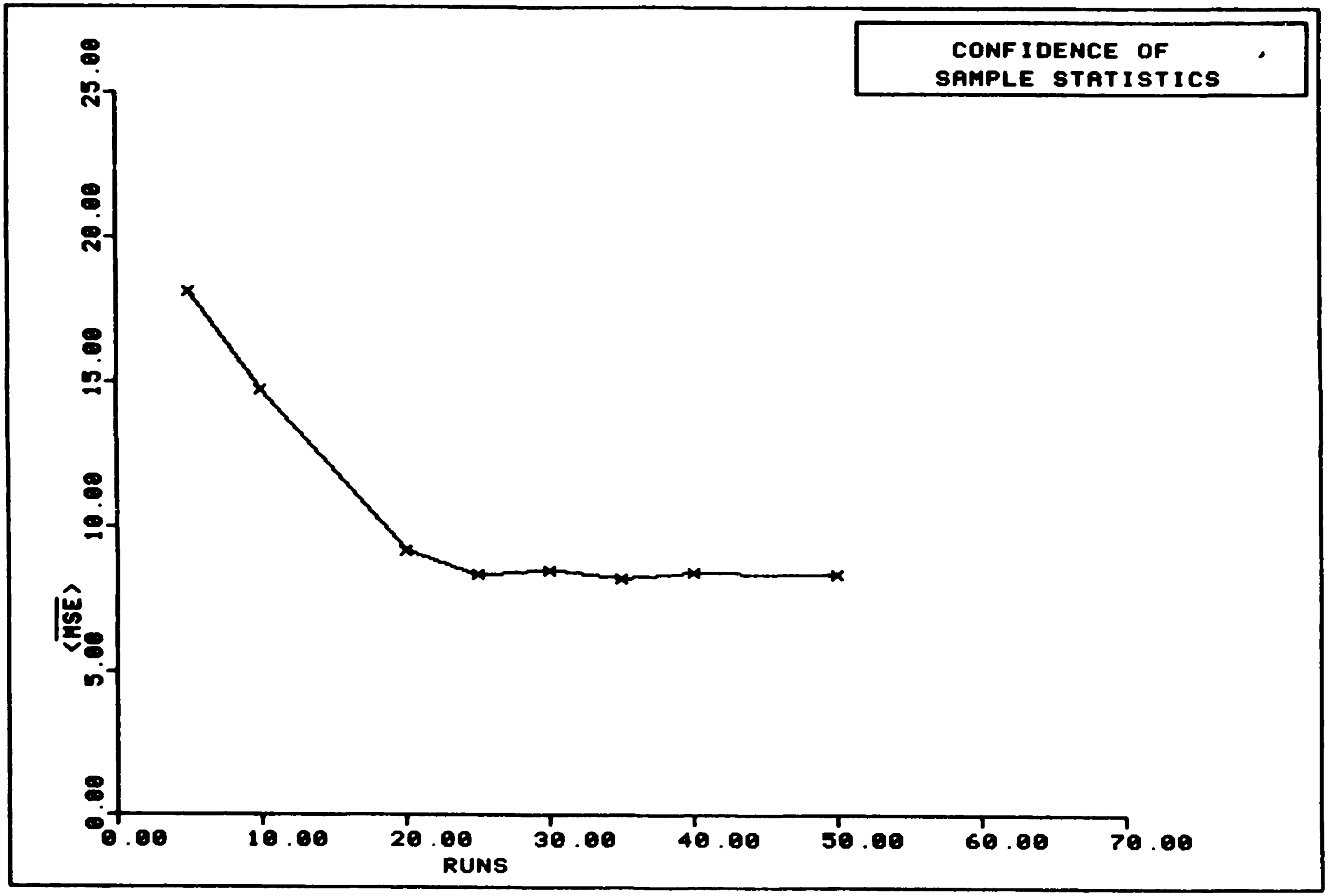


FIGURE 17. Number of Runs

rate increases. Since the rate parameters are derived from a uniform, (0, 1), distribution of random numbers and the relation of equation (57), an average rate parameter of .6 corresponds to the expected value of an ensemble of rate parameter derived from a uniform distribution of random variates. Therefore, a range of values for the average signal rate of (.6 - .7) was used when comparing ensemble average mean-square error, ensemble average state error and average probability of error.

(2) Performance Measures. Estimator performance is expressed in four ways. The first is actual comparison between the original state sequence and estimated state sequence. This comparison will demonstrate the performance of the estimator for a specific signal with respect to providing an accurate reconstruction. The second and third method of expressing performance is from the ensemble average mean-square error and the ensemble average state error. The average state error is an ensemble average of the difference between the estimated state at time/sample K and the actual state at that time. These methods depict the estimator's performance over an entire ensemble of signals within a given set. The fifth method will be defined as the average probability of error, $\langle PE \rangle$. For this study, the average probability of error is simply defined as the ratio of the average number of error estimates to the total number of

estimates made. This measure tends to be sensitive and misleading. Simply put, this measure tells how often the estimator was in error on an absolute scale (Ref 11). The $\langle PE \rangle$ is sensitive to all parameter changes, particularly the average rate used to generate the ensemble of signals. A lower average rate parameter, $\langle RATE \rangle$ produces a lower $\langle PE \rangle$ The measure is misleading in that for an estimate of 640 samples (as will be shown), there is a point after which trying to improve this measure results in no noticeable change in the estimated sequence (assuming an accurate reconstruction has been achieved). For example, a $\langle PE \rangle$ equal to .20 appeared to indicate the estimator was able to reconstruct each signal of the ensemble accurately. If parameters were changed to improve this figure, .2, the estimator output will improve on an estimate-by-estimate basis, however, graphic comparison between the original sequence for a given signal reveals no appreciable enhancement.

A different measure has beer. defined to replace the conventional signal-to-noise ratio (SNR) reference. The new reference is the average threshold crossing ratio, $\langle ZCR \rangle$. The SNR term has little meaning based on the assumptions and characteristics presented earlier. The $\langle ZCR \rangle$, however, is a useful reference for quantifying estimator performance. This ratio is defined as the average total number of pulses detected to the number of signal pulses per frame. Since the number of signal pulses is always one, this figure expresses the degree to which the sample value (pulse position) has been masked or degraded by anamolous pulses. Referring to Figure 5, the $\langle \mathrm{ZCR} \rangle$ provides the average number of dots per sample time. Performance comparisons will generally be made with reference to a $\langle \mathrm{ZCR} \rangle$ measure. The terms $\langle \mathrm{ZCR} \rangle$, $\langle \mathrm{PE} \rangle$, $\langle \mathrm{RATE} \rangle$ appear on each performance plot for reference. Additionally, the average difference between the assumed starting state and the actual starting state, $\langle \mathrm{DIFF} \rangle$ is also provided.

(3) Estimator Performance. Each sample signal was tested using the appropriate models (58), (59), or (60). Initially, the lower sample rate of f_S=50 S/S was used along with a decision depth, J_o=5. This sample rate and decision depth proved adequate for relatively low $\langle \mathrm{ZCR} \rangle$. Adequate refers to the estimator's ability to provide a reconstructed version of the original sequence that retains a recognizable shape and has a minimum of jump discontinuities (i.e., state transitions of greater than ± 1). NOTE: In this section, signal #6 is used to present the performance of the estimator. The other signals will be introduced when the estimator performance differs significantly from that for signal #6. NOTE: The initial simulations were conducted using a zero-mean Guassian channel model. It became apparent that, assuming a positive threshold, capable of detecting a

positive going pulse, the largest number of threshold crossings per frame, due to noise approaches a limit of 16 only as the threshold gets close to zero. It was realized that in order to produce a greater number of anamolous pulses, a nonzero mean channel model must be considered. The Poisson channel model appeared most promising at first. Problems arose, however, in the implementation of the Poisson model and it was abandoned for a nonzero mean Gaussian density for high count rates (Ref 11), therefore, no loss in accuracy was anticipated nor observed. In this manner also, the metric of equation (53) essentially remains unchanged. The problems associated with the Poisson channel development are presented in Appendix A.

The ensemble average error and the ensemble average mean-square error are shown in Figures 18-23 for $\langle ZCR \rangle$ values up to 23 (average A=22). The mean-square error and state error graphs provide information regarding the algorithm's ability to track each ensemble. Each graph shows the catchup time and associated error and the transition time and associated error. The catchup time and error is caused by the difference between the actual starting state and the assumed starting state. The transition time error is caused by the inability of the algorithm to track the fastest changes in the original state sequence and/or the inability of the algorithm to follow the transition of the signal near the five second point. For purposes of brevity, these errors will be referred to as transient error and transition error.

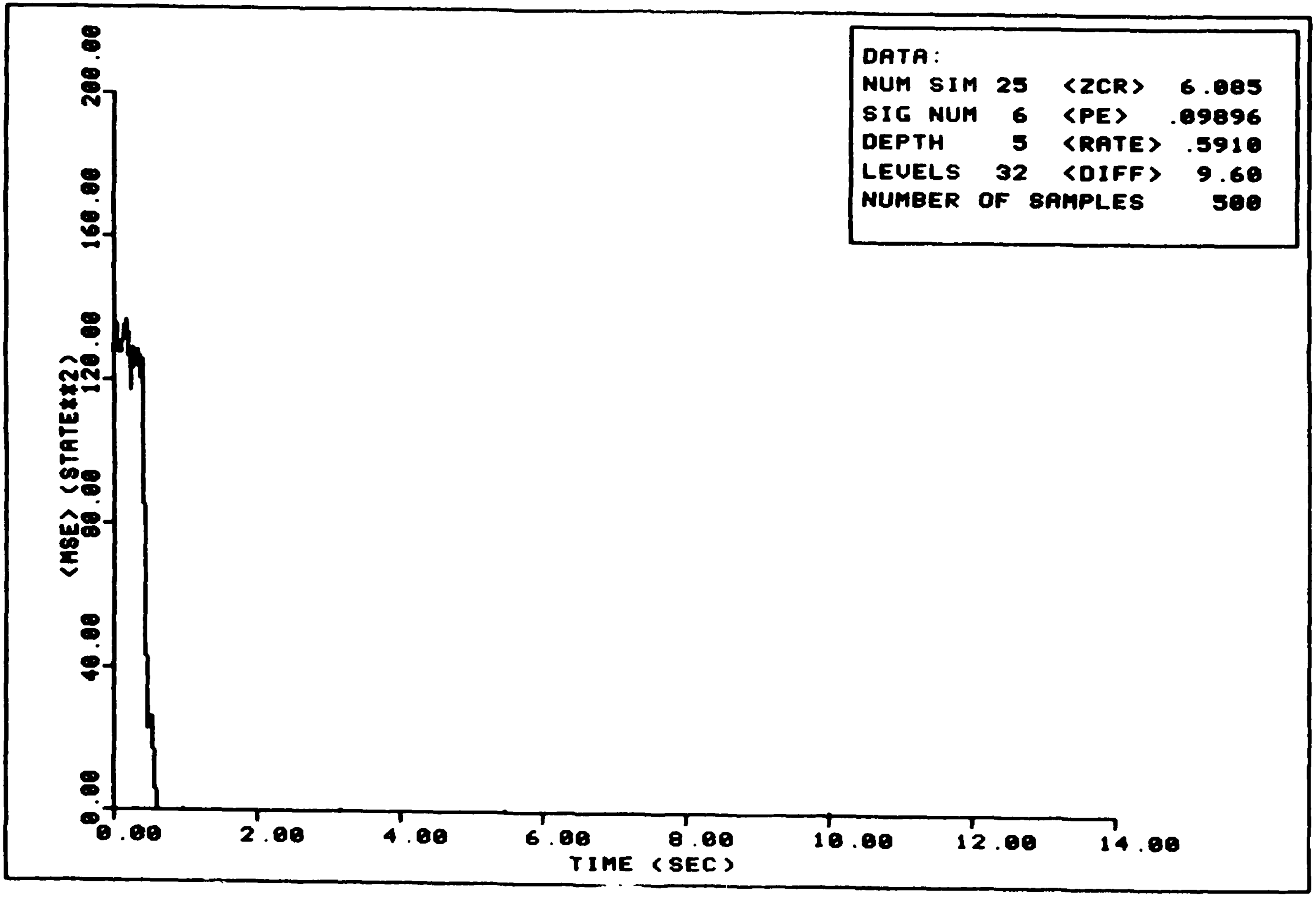


FIGURE 18. Ensemble Average MSE Performance

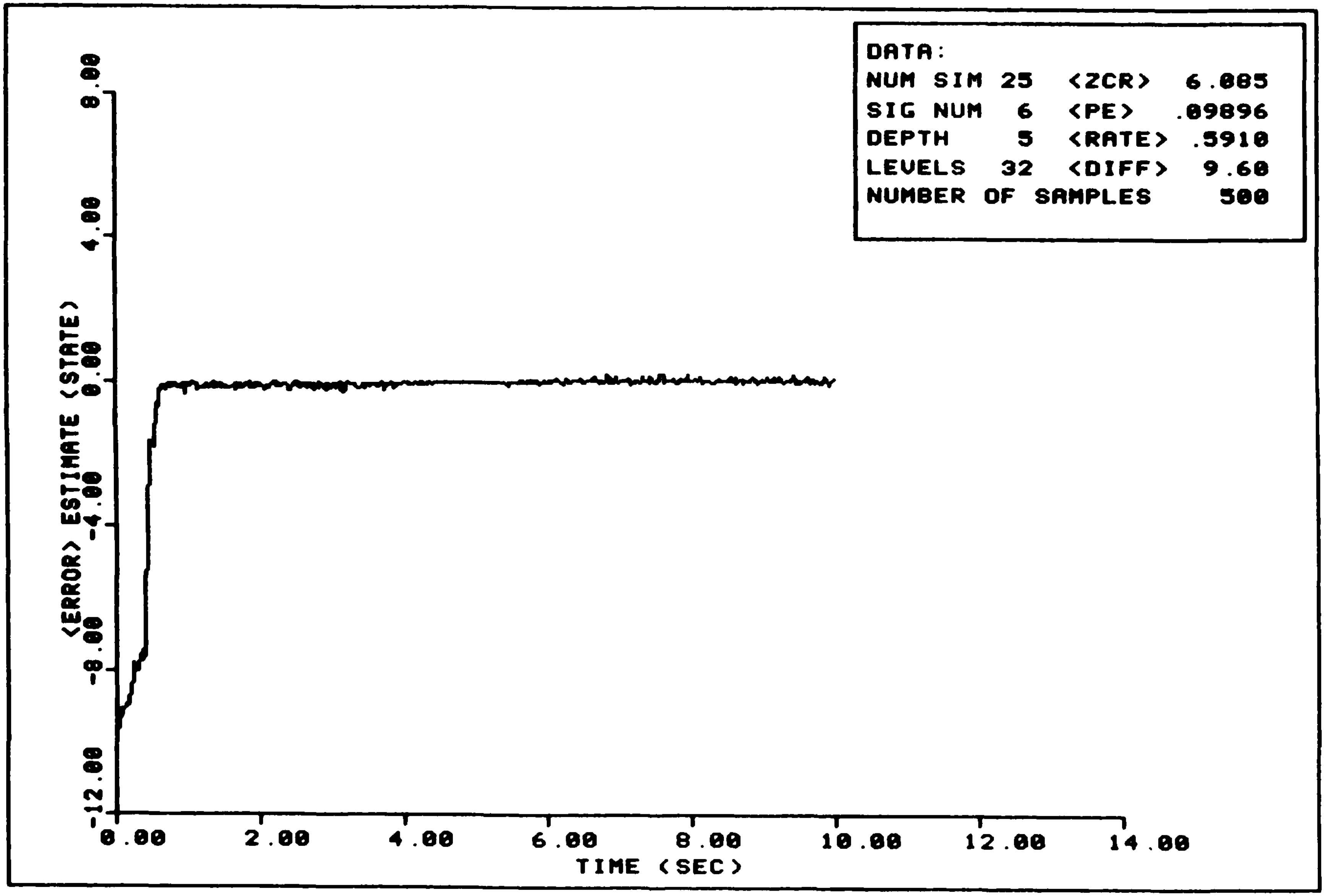


FIGURE 19. Ensemble Average State Error Performance

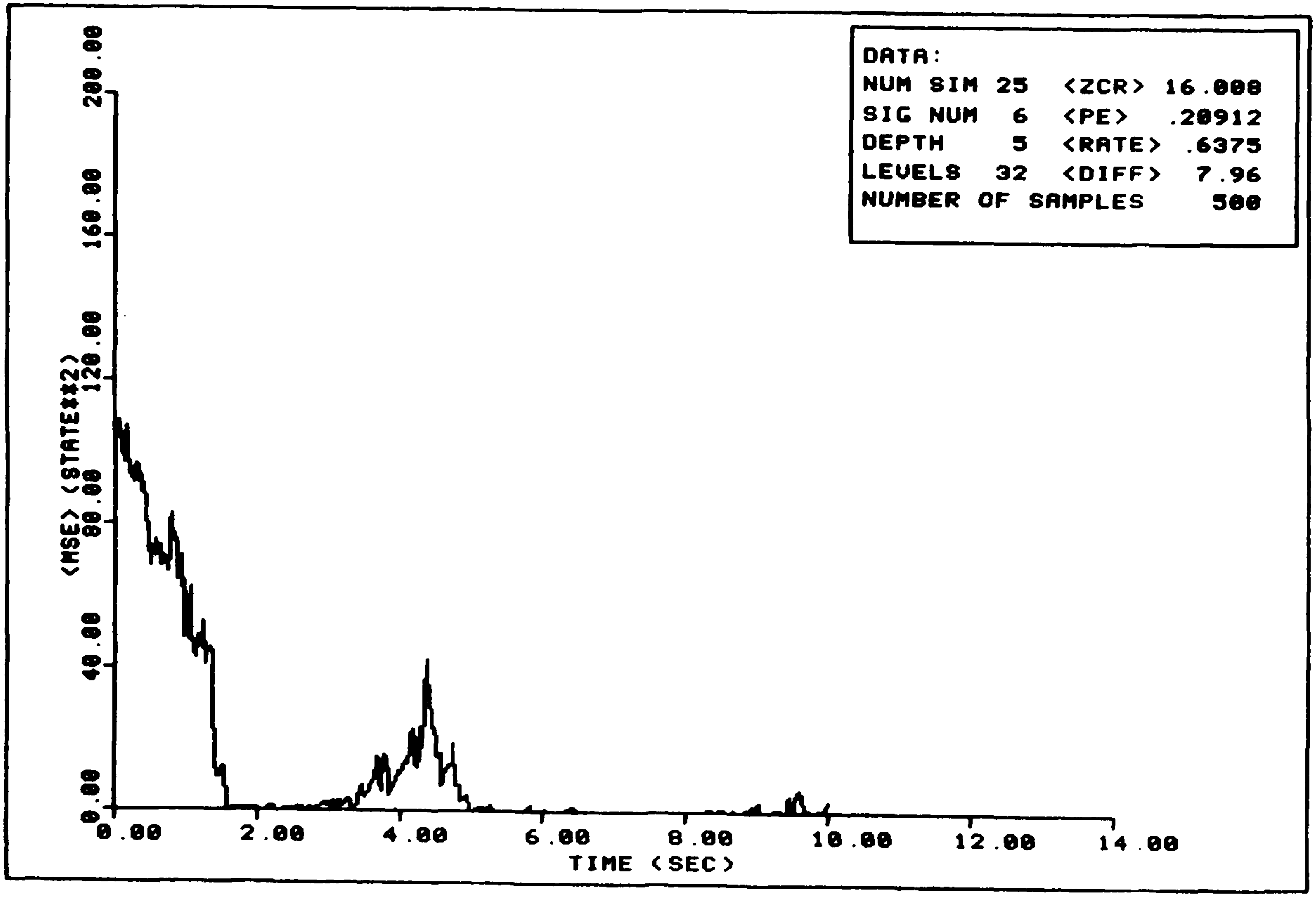


FIGURE 20. Ensemble Average MSE Performance

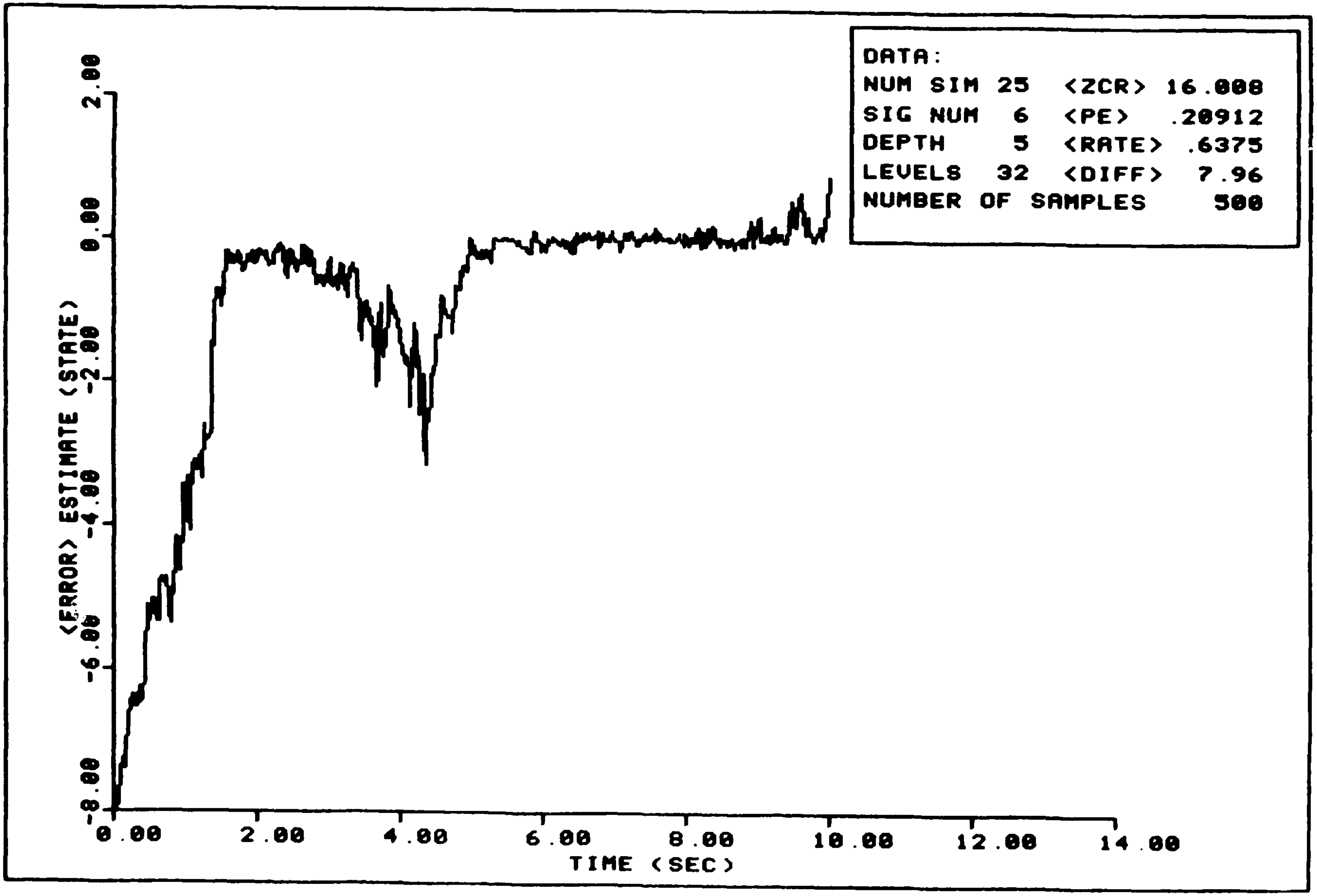


FIGURE 21. Ensemble Average State Error Performance

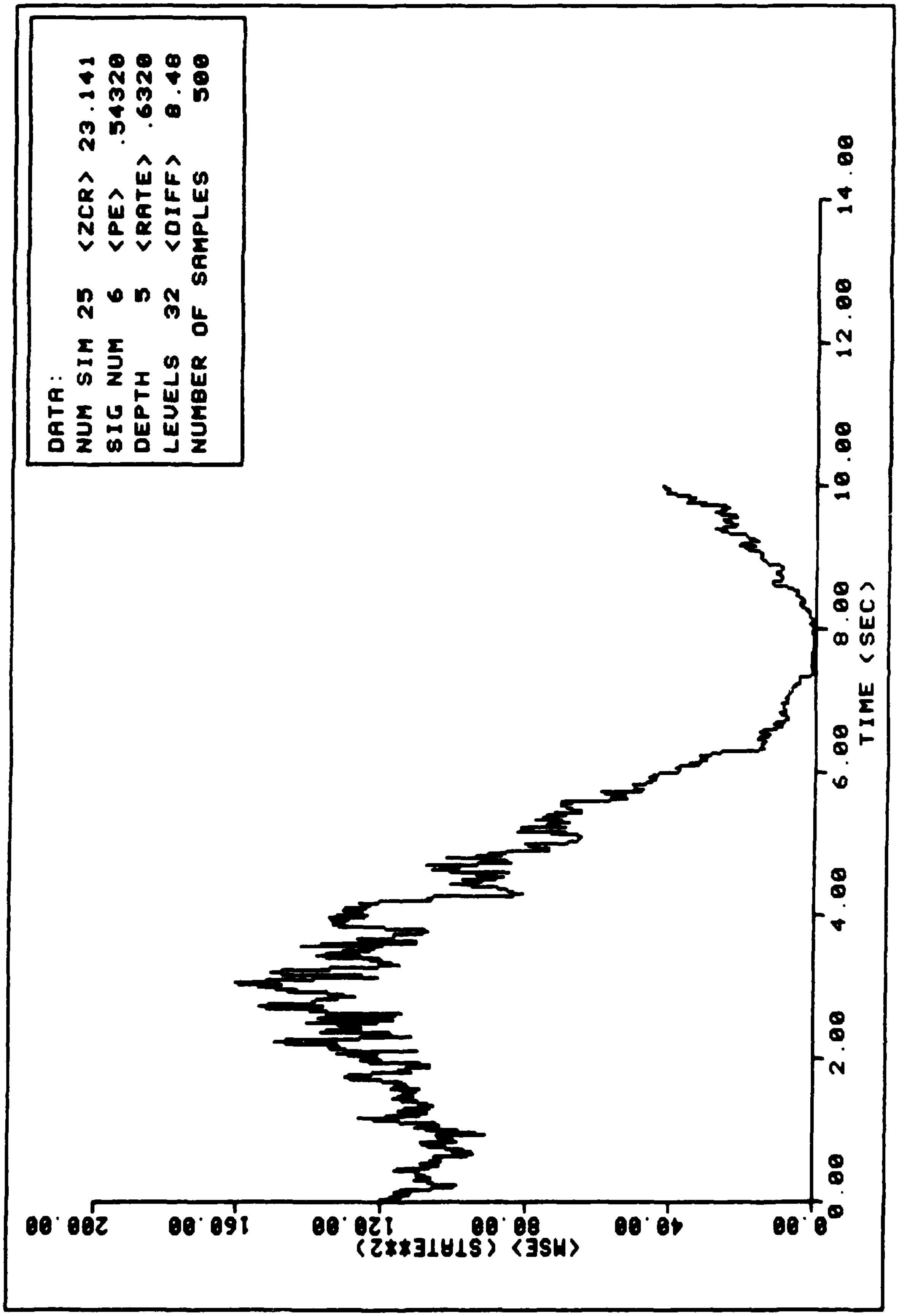


FIGURE 22. Ensemble Average MSE Performance

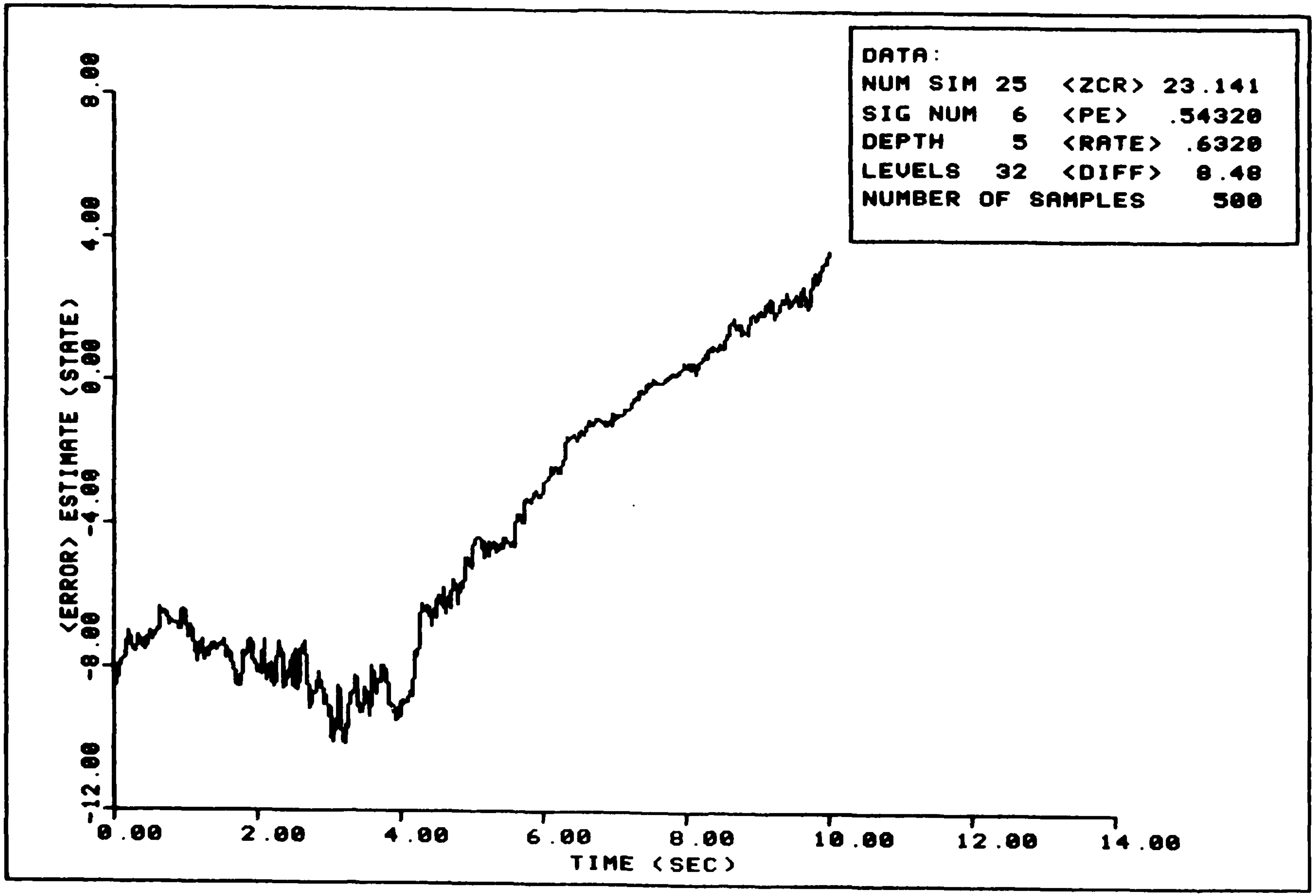


FIGURE 23. Ensemble Average State Error Performance

For low $\langle ZCR \rangle$ as shown in Figures 18 and 19, only the transient error is evident. The ensemble average mean-square error graph provides little information for low $\langle ZCR \rangle$ values. The ensemble average state error graphs, however, show more completely the trends in algorithm performance. The state error graphs reflect periods of relative stability that exist generally between the transient and transition error times. As will be seen, as the $\langle ZCR \rangle$ value increases, the transient error occurs and transition times will lengthen, thus reducing the time of estimator stability.

A sample state estimation is given in Figures 24-32. The limits on performance may be realized by comparing these six figures. As can be seen, the estimator provides an accurate reconstruction of the original state sequence for low $\langle ZCR \rangle$. As the number of threshold crossings per frame is increased, two factors are evident. First, the transient error increases because the algorithm takes longer to catch up with the original sequence as in comparing Figures 24 and 27. Second, the transition error begins to appear for high values of the rate parameter, r. This error begins to appear for $\langle ZCR \rangle$ as low as eight, but does not become significant until the $\langle ZCR \rangle$ reaches 15. Figures 29-32 show significant transition error. As the number of threshold crossings is increased, the transition error occurs for lower values of r and the transition error region widens for

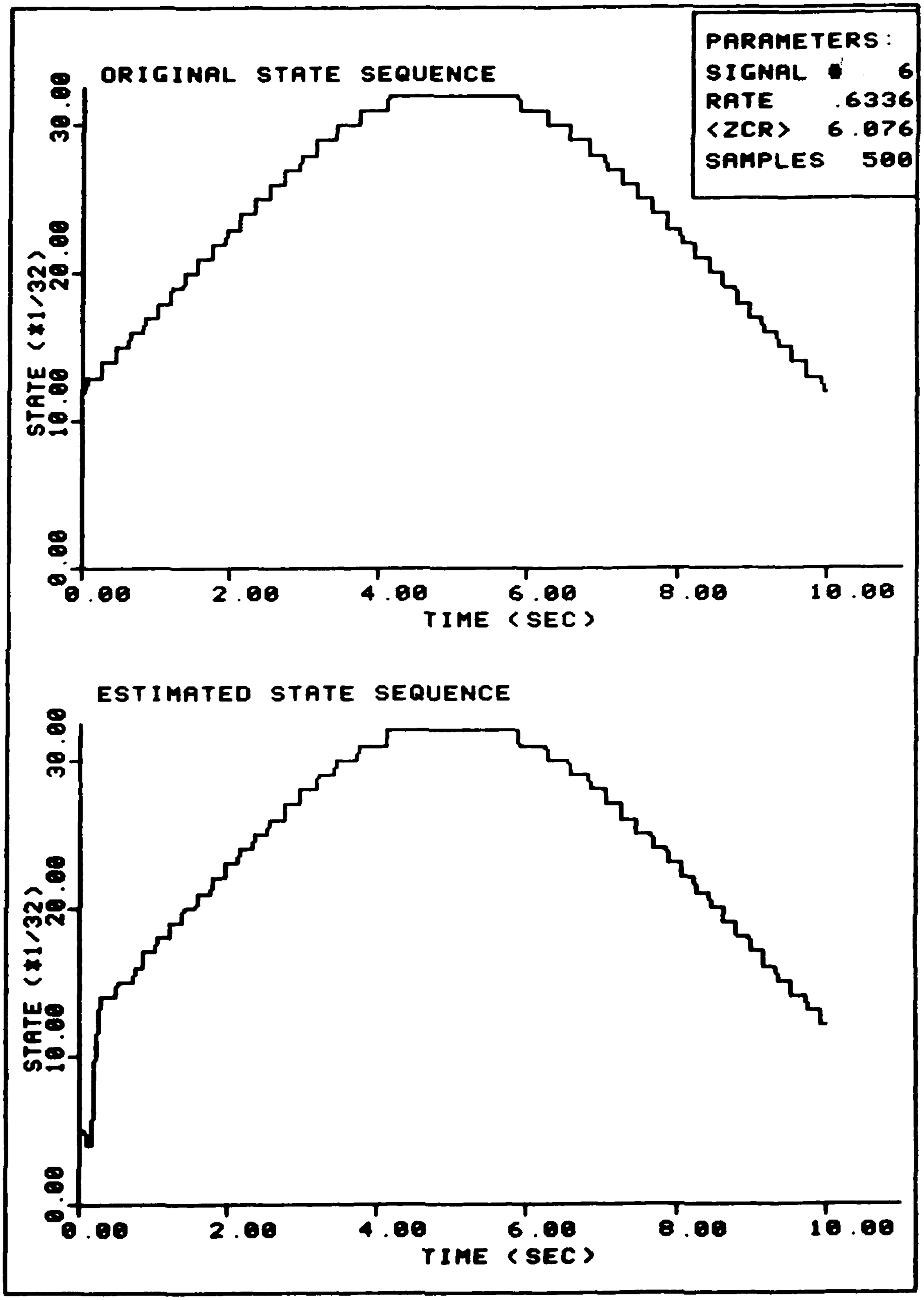


FIGURE 24. Sample Estimation - Depth=5

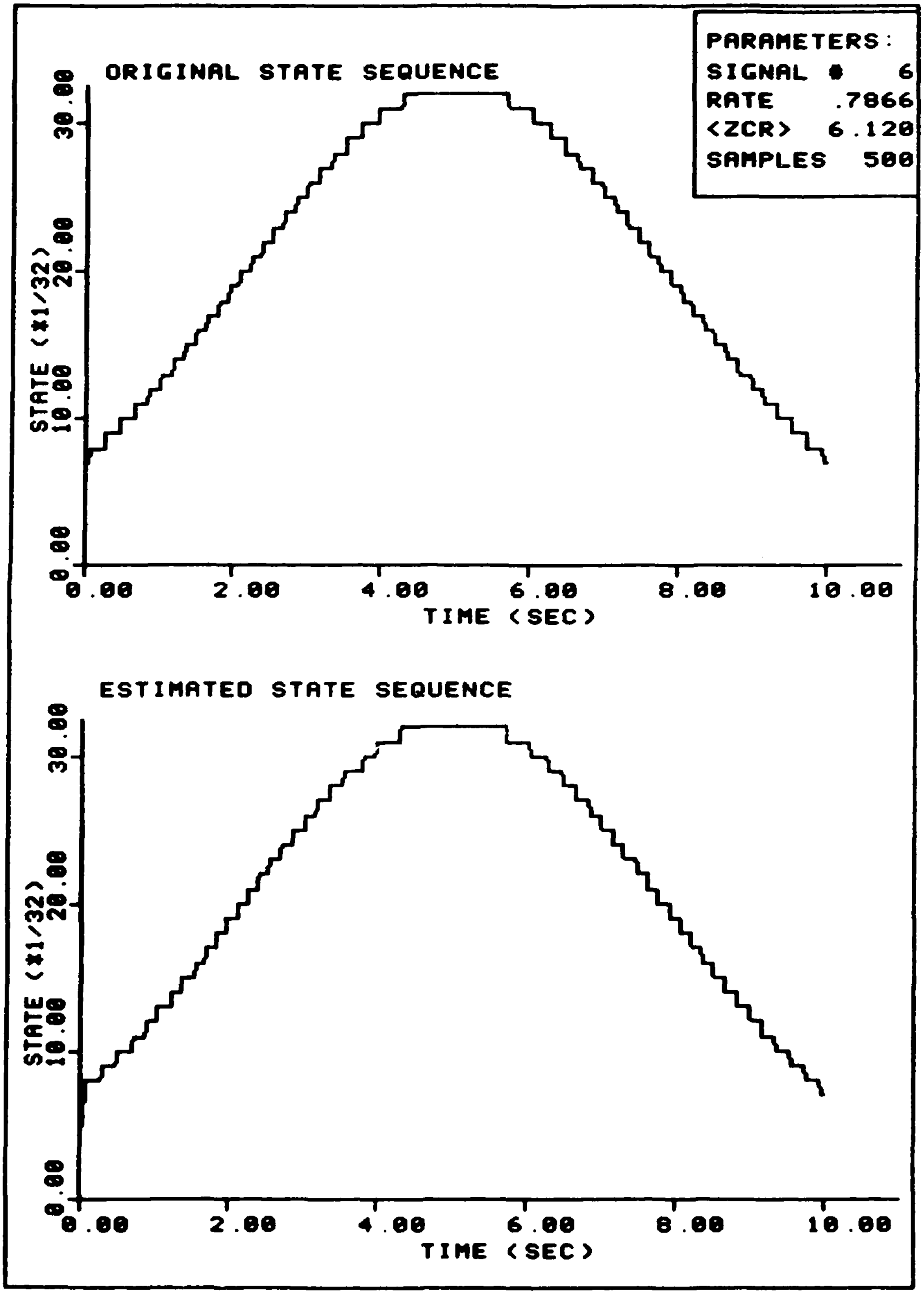


FIGURE 25. Sample Estimation - Depth=5

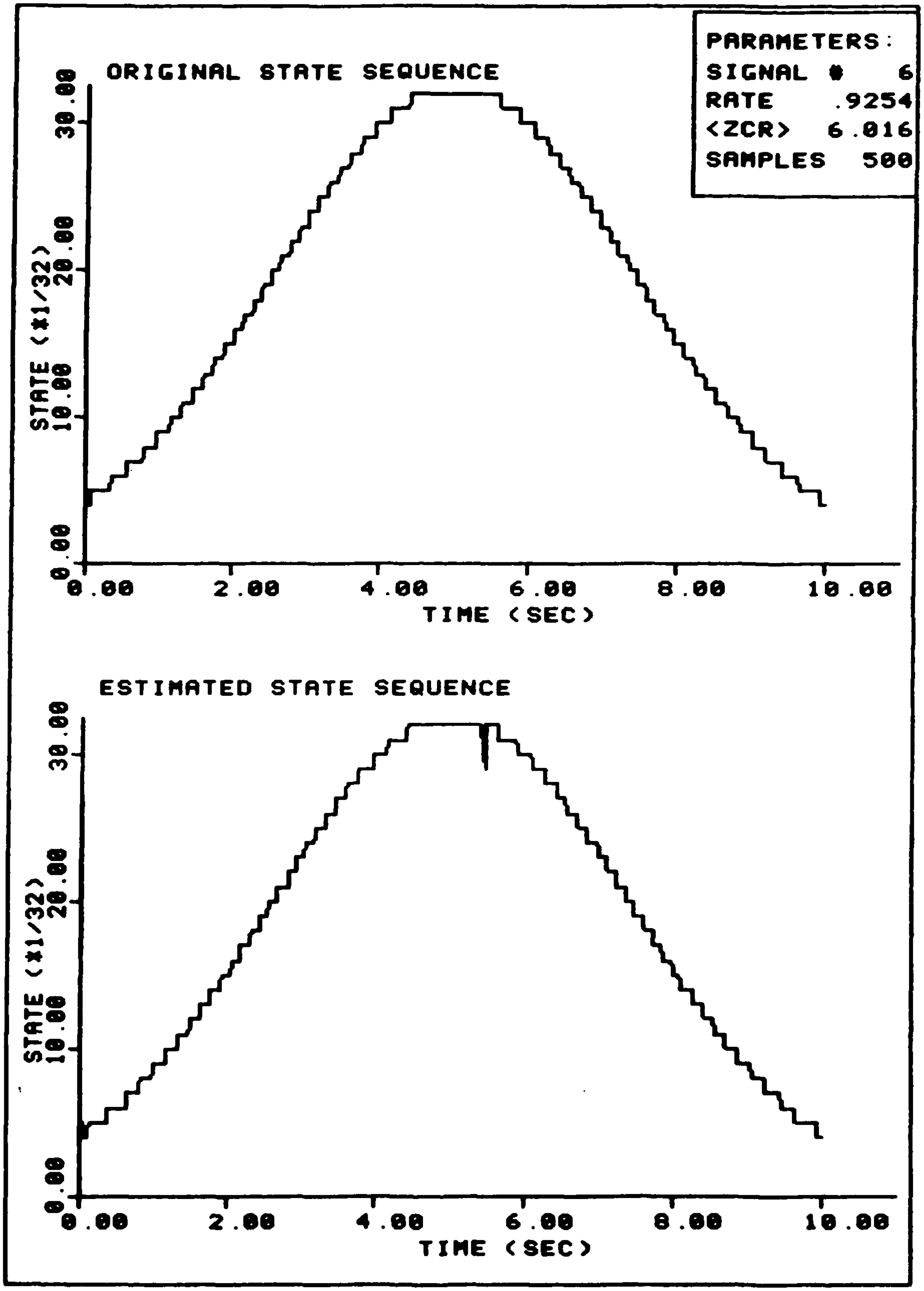


FIGURE 26. Sample Estimation - Depth=5

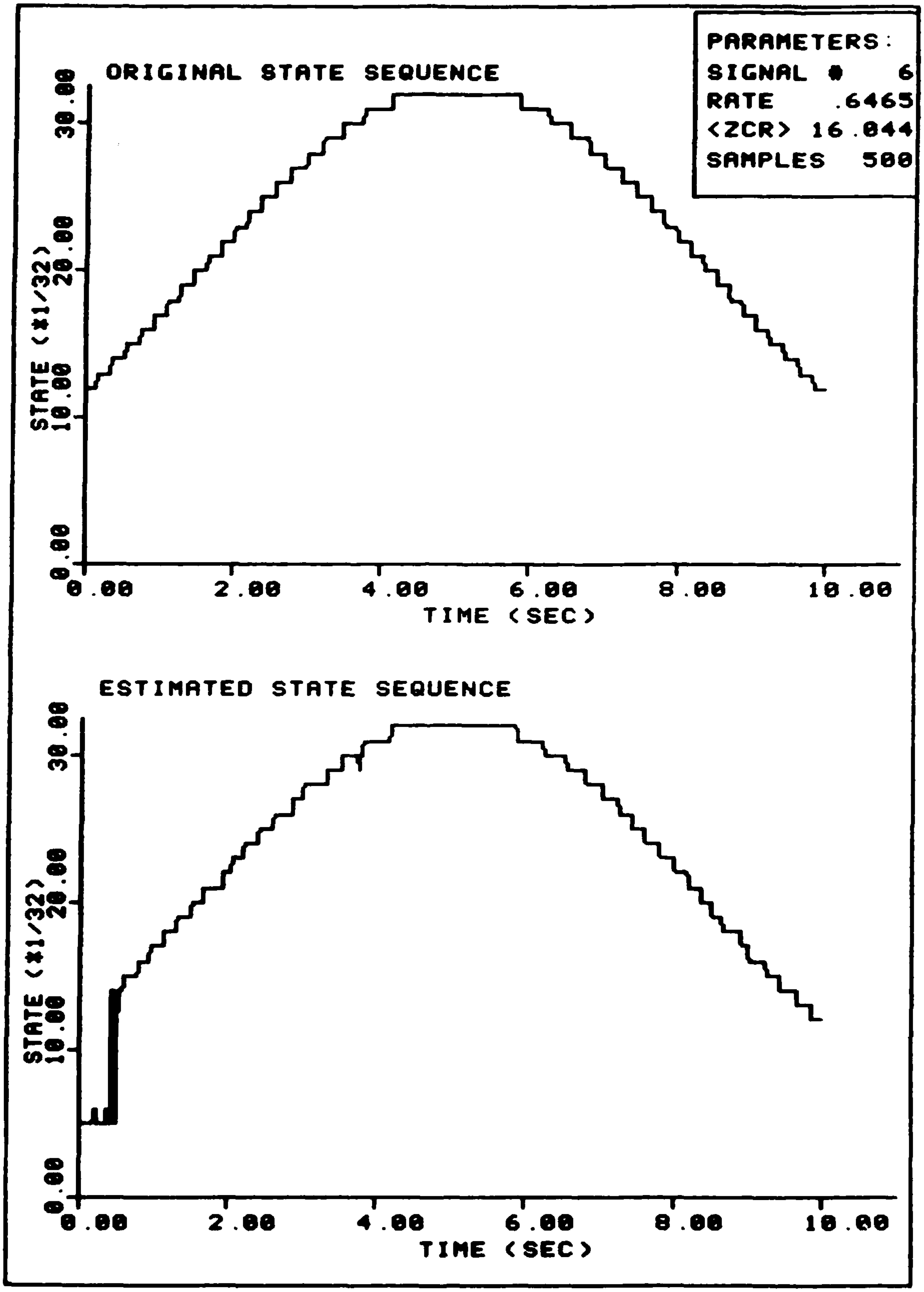


FIGURE 27. Sample Estimation - Depth=5

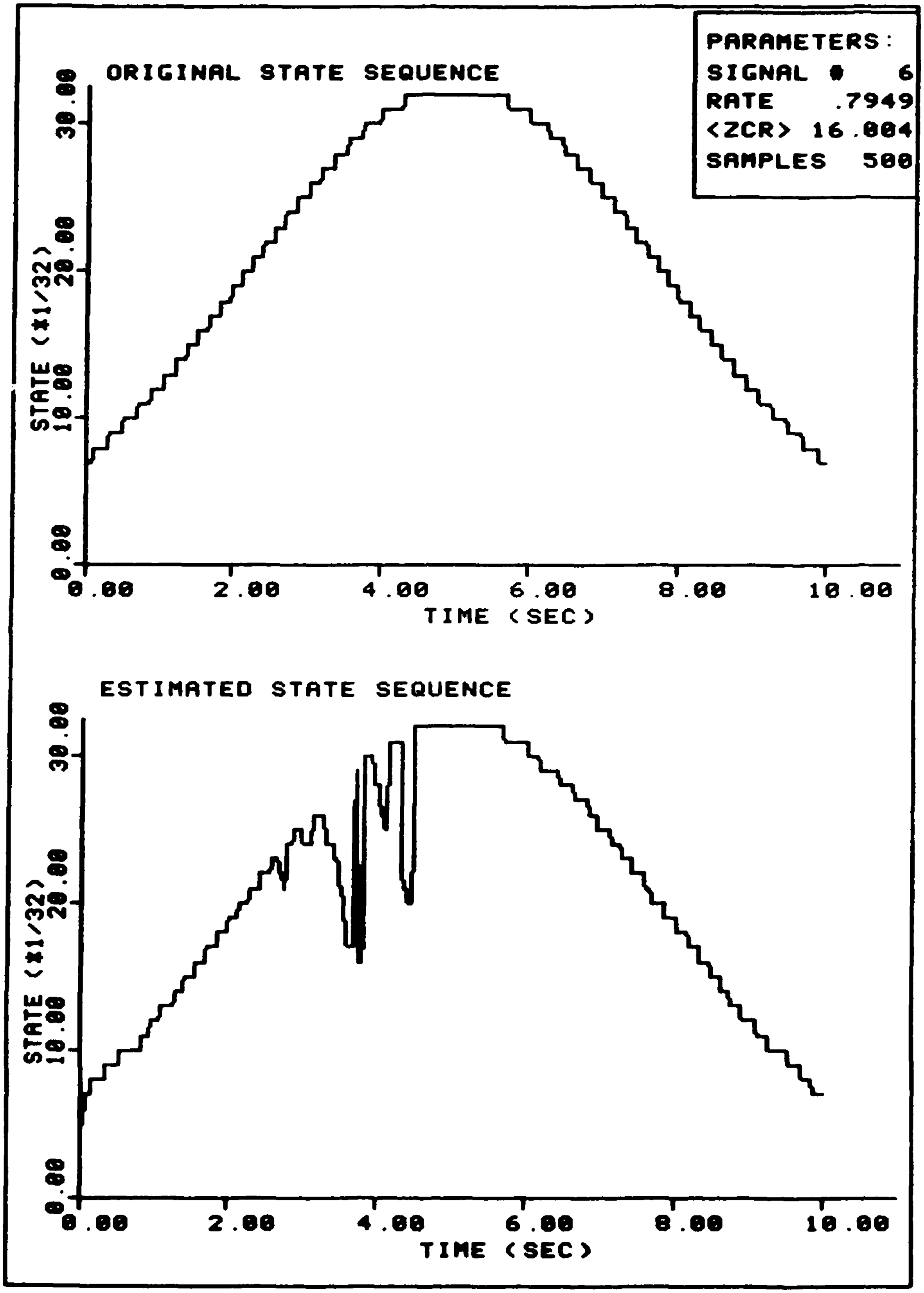


FIGURE 28. Sample Estimation - Depth=5

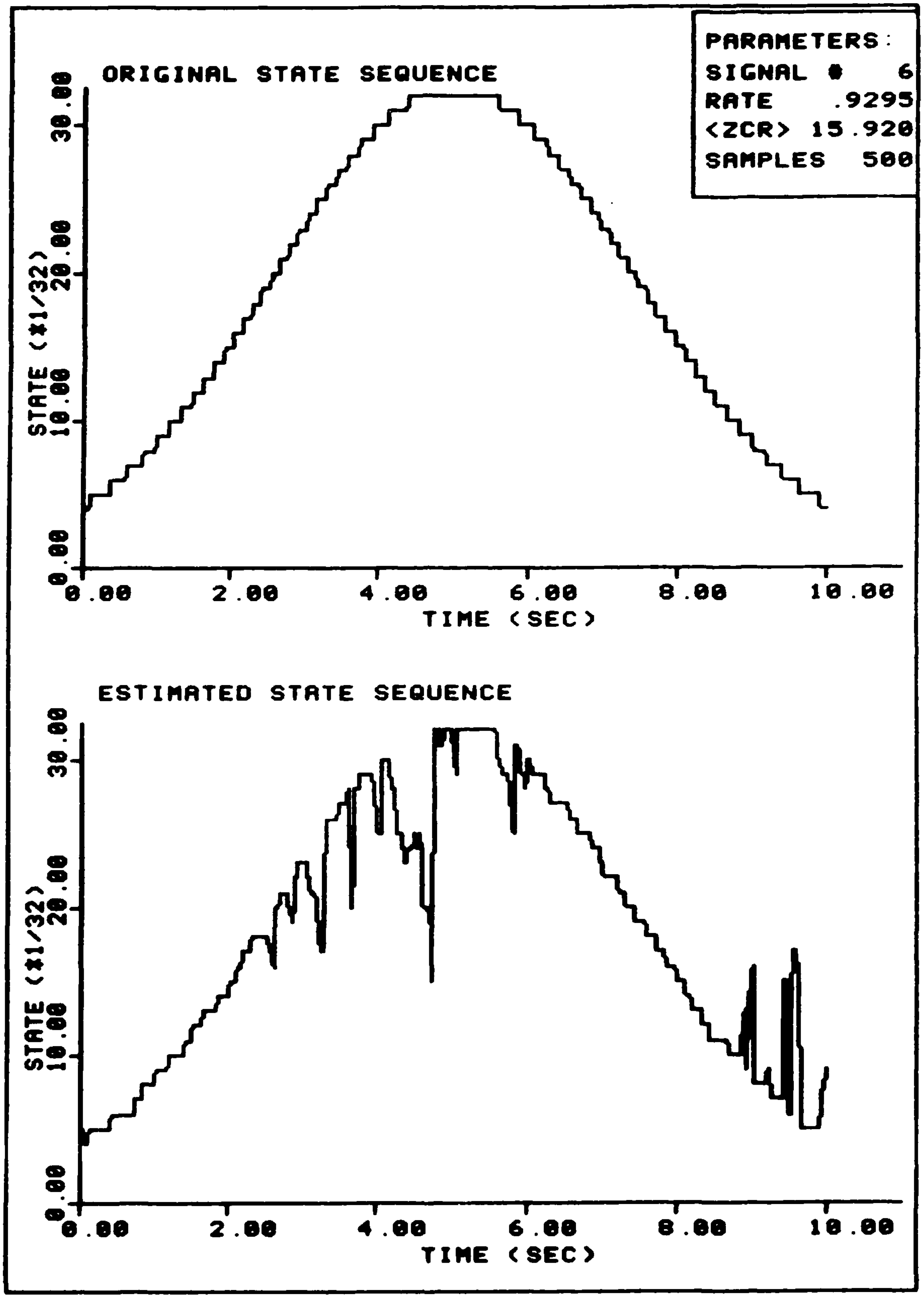


FIGURE 29. Sample Estimation - Depth=5

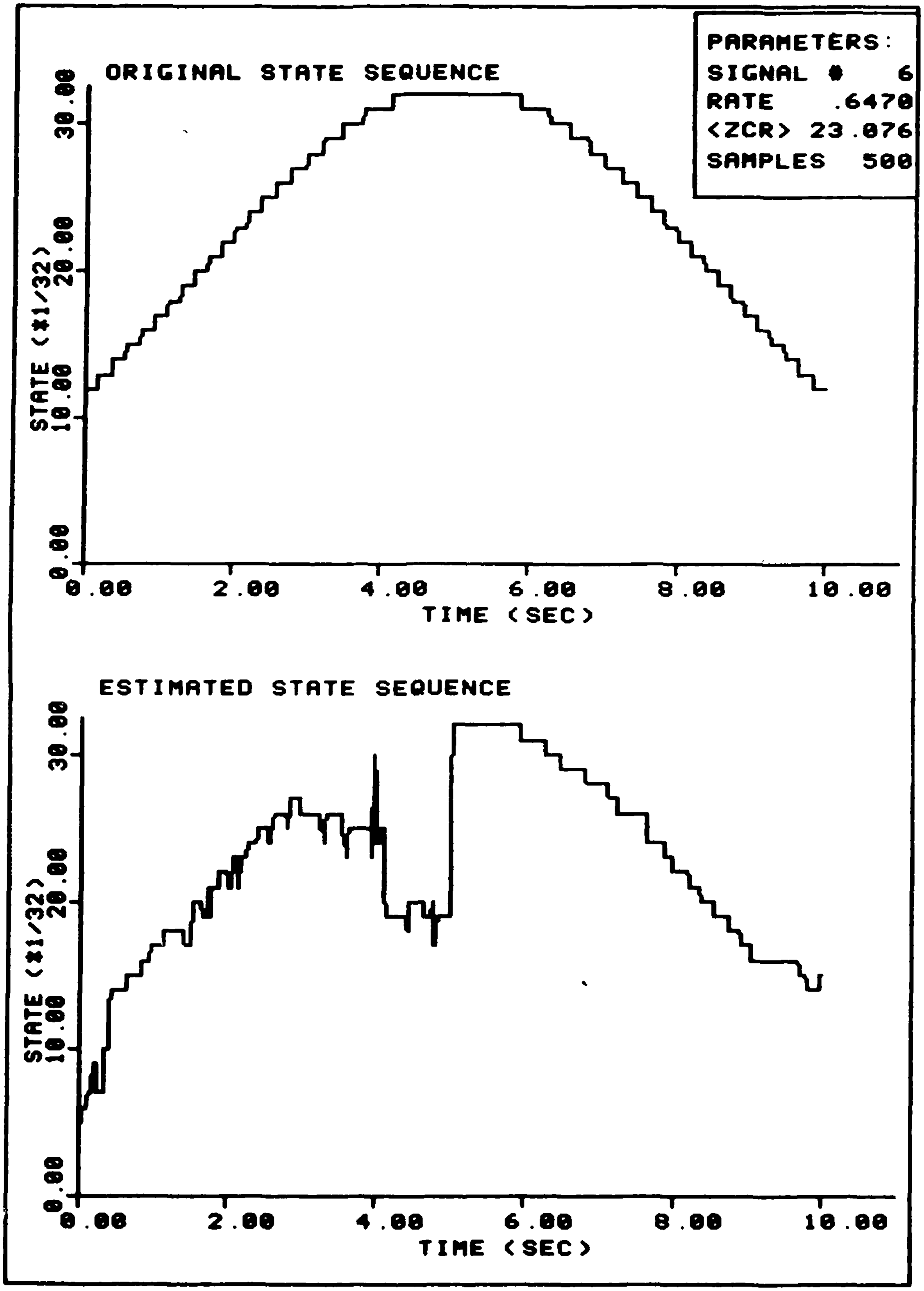


FIGURE 30. Sample Estimation - Depth=5

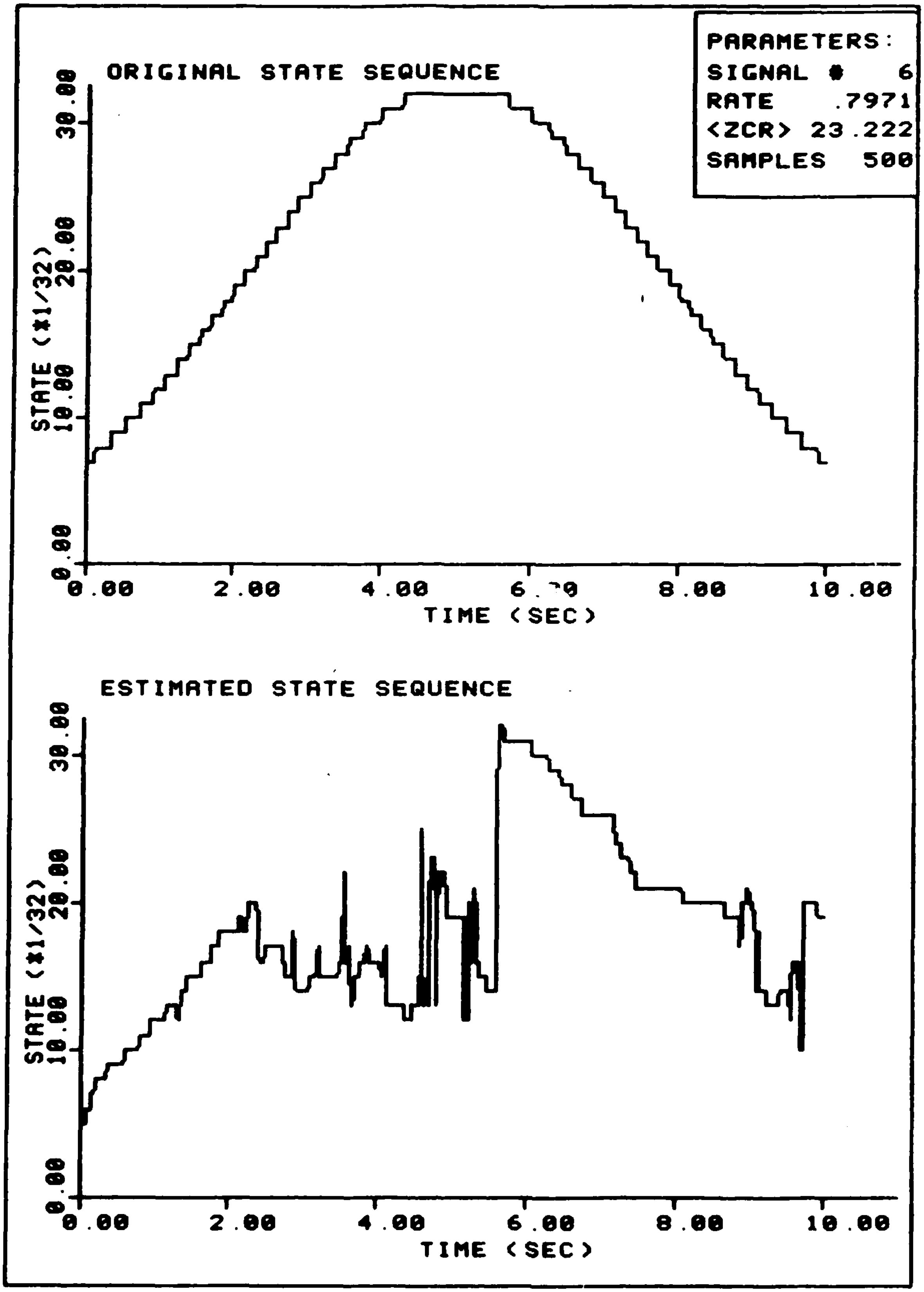


FIGURE 31. Samɥle Estimation - Depth=5

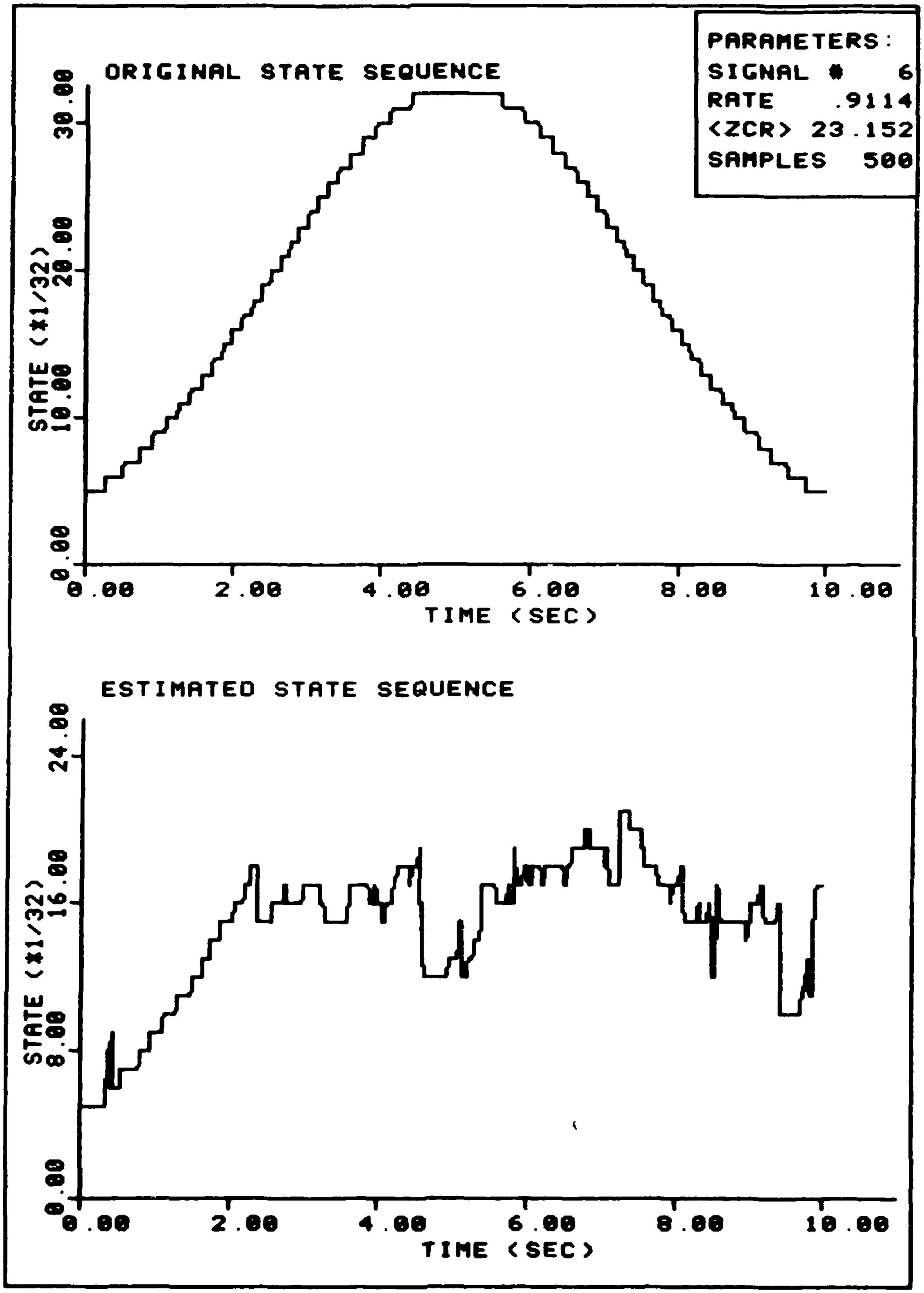


FIGURE 32. Sample Estimation - Depth = 5

higher values of r. Figures 27, 28 and 29 show the tracking problem for a $\langle \mathrm{ZCR} \rangle$ of approximately 16. As the $\langle \mathrm{ZCR} \rangle$ is increased further, the transient and transition errors merge. The tracking problem is depicted in Figures 30-32 for $\langle \mathrm{ZCR} \rangle = 23$ and the associated average mean-square error and average state error in Figures 22 and 23. The effect of increasing $\langle \mathrm{ZCR} \rangle$ is evident by comparing Figures 24-32.

The estimator performance in Figures 24-32 may be explained in terms of the two quantities of equation (48) that constitute the branch metrics, and the two types of errors described earlier (transient and transition). It was explained earlier that the transient error is caused by the difference between the assumed starting state and the actual starting state. This error becomes more significant with increasing $\langle \mathrm{ZCR} \rangle$. However, this error may be reduced or even eliminated by providing more accurate starting state information. Note the lack of significant transient error in Figures 29 and 31 in spite of the obvious problems the algorithm has in tracking the sequence. For these sequences, the assumed starting state is more accurate.

The appearance of the transition error may be attributed to parameters of equation (48), the branch metric. For low $\langle \mathrm{ZCR} \rangle$ values, the distance term of equation (48) as given by equation (51), provides the distinguishing parameter to each branch. That is, for low $\langle \mathrm{ZCR} \rangle$, the algorithm is able to use the distance measure to a greater extent in prɔviding the shortest path through the trellis. In this manner, the algorithm is able to track rapid changes (within the ±1 state limit) in the signal sequence, even when the $\overline{P}$ matrix indicates a low probability of such an occurrence. How the $\overline{P}$ matrix provides this indication is explained next.

As the $\langle ZCR \rangle$ increases, the algorithm must rely on the priors, the transition probability matrix, to provide the distinguishing parameter in identifying the shortest path. Recall that the values of the $\overline{P}$ matrix are not time varying but are dependent upon the current occupied state (i.e., current population). As the que develops or the population grows, the expression for the birth-death coefficients in equation (32) yields a greater probability for a minus one state transition (a death) or no state transition, than for a plus one state transition (a birth). When the algorithm relies more heavily on the priors to compute the shortest path, this path will reflect the $\overline{P}$ matrix as defined by the birth-death coefficients.

This indeed does occur in Figures 30-32. The change-over point in the $\overline{P}$ matrix for when a death is more likely than a birth corresponds to approximately state 16. The estimator has some difficulty in overcoming this "inertia" of the $\overline{P}$ matrix, particularly in Figure 32. In Figure 31, the estimator cannot track the sequence during the up-slope side, but does relatively well tracking the <u>same</u> rate of change on the down-slope side. The behavior of the signal in Figure 31 for t $\geq$5.0 "fits" the $\bar{P}$ matrix. That is, for high current state values, the probability of a death is much greater than the probability of a birth. Therefore, since the algorithm relies more on the transition probabilities, and since the message behavior matches the $\bar{P}$ matrix, the estimated sequence appears accurate.

In this manner, the performance is somewhat predictable for high $\langle ZCR \rangle$ values. For each signal (#3, 6, 7, 8, 9, 10), the region in which the transition error first occurs is predictable based on a comparison between signal behavior and the $\bar{P}$ matrix. For example, for signal #6, this region is on the up-slope side as depicted in Figure 31. For signal #7, this region is on the down-slope side as depicted in Figure 103. This mismatch for signal #7 is where the $\bar{P}$ matrix indicates a greater probability of a birth than a death within the que, but the signal is experiencing its greatest down-slope rate. Interestingly, for signal #9, there are two mismatch regions: one on the up-slope side similar to signal #6, and one on the down-slope side similar to signal #7. Similar regions exist for signal #3 and signal #10.

The relative performance of the estimator may be seen in Figure 33 with respect to the average probability of error measure, $\langle PE \rangle$. Figure 33 demonstrates the sensitivity of this measure, particularly for $\langle ZCR \rangle$ value above

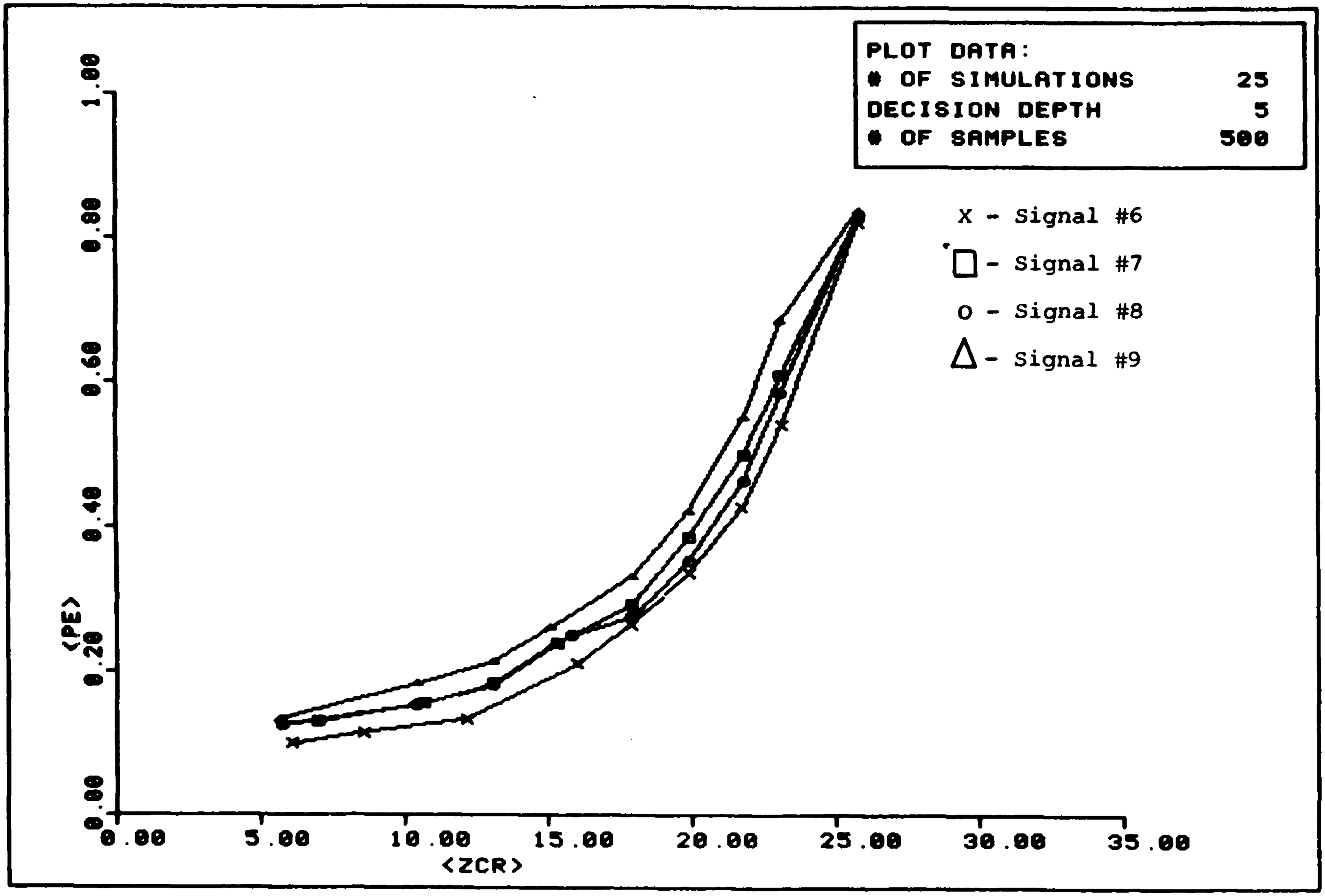


FIGURE 33. Performance Comparison

16. NOTE: This was true for all signals. At this point in the simulation, it became apparent that an accurate reconstruction of the original state sequence could be guaranteed for $\langle PE \rangle \lesssim .20$ over the entire ensemble of rate parameters. Marginal reconstruction is possible for $.20 \lesssim \langle PE \rangle \lesssim .28$ and only immaginative reconstruction is possible for $\langle PE \rangle$ $> .28$. This range of values for $\langle PE \rangle$ will be reinforced as the various parameters (T, J_o) are changed.

The hypothesis presented earlier calls for a minimum decision depth based on the assumed starting state. The sample rate hypothesis of equation (61) suggests that increasing the sample rate (increasing K in equation (61)) will result in better performance. Computer resources restricts the values of K that may be used. Therefore, the value of K was increased to 2, providing a sample rate of 64 S/S. This value provides a hypothesized minimum sample rate for all rate parameters which will be used for the remainder of the simulation.

Before the new sample rate was increased, the effect of increasing the decision depth was examined. The decision depth J_o was increased incrementally from 5 to 65. The effect on $\langle PE \rangle$ is seen in Figure 38. Once again, the range of $\langle PE \rangle$ that guaranteed an accurate reconstruction for the entire ensemble was $\langle PE \rangle \lesssim .20$. The effect of increasing the decision depth is to lower the $\langle PE \rangle$. The figure shows that increasing the decision depth beyond 20 reduces the $\langle PE \rangle$ only fractionally. Recalling that the sample rate provides the memory for the algorithm, Figure 38 demonstrates that no benefit is derived in delaying a decision (output an estimate) longer than the extent of the memory supplied by the priors.

Performance for $\langle ZCR \rangle > 23$ deteriorates rapidly. Figures 34 and 35 demonstrate the extent of the degraded performance when the $\langle ZCR \rangle$ is increased to 26. The figures show that the $\langle MSE \rangle$ tends to take off for $\langle ZCR \rangle >$ 23, but tends to stay flat for $\langle ZCR \rangle \simeq 23$ (Figure 22). This performance suggests that steps taken to improve performance, within the limits specified earlier, would have a more noticeable effect for a $\langle ZCR \rangle \simeq 23$. For these reasons, the effect of increasing the sample rate and decision depth was observed for $\langle ZCR \rangle \simeq 23$, primarily. These results are presented next and a comparison is made to the performance under the conditions of the lower sample rate and decision depth.

The performance of the estimator, under the new conditions is shown in Figures 36 and 37. A comparison of Figures 36 and 37 to Figures 22 and 23 reveals that increasing the sample rate alone, has a slight effect on the $\langle PE \rangle$, and $\langle MSE \rangle$. In this case, the higher sample rate has provided more memory, however, the advantage of this memory is minimized by forcing early decisions (J_o=5). Simulations using longer decision depths were conducted to produce the

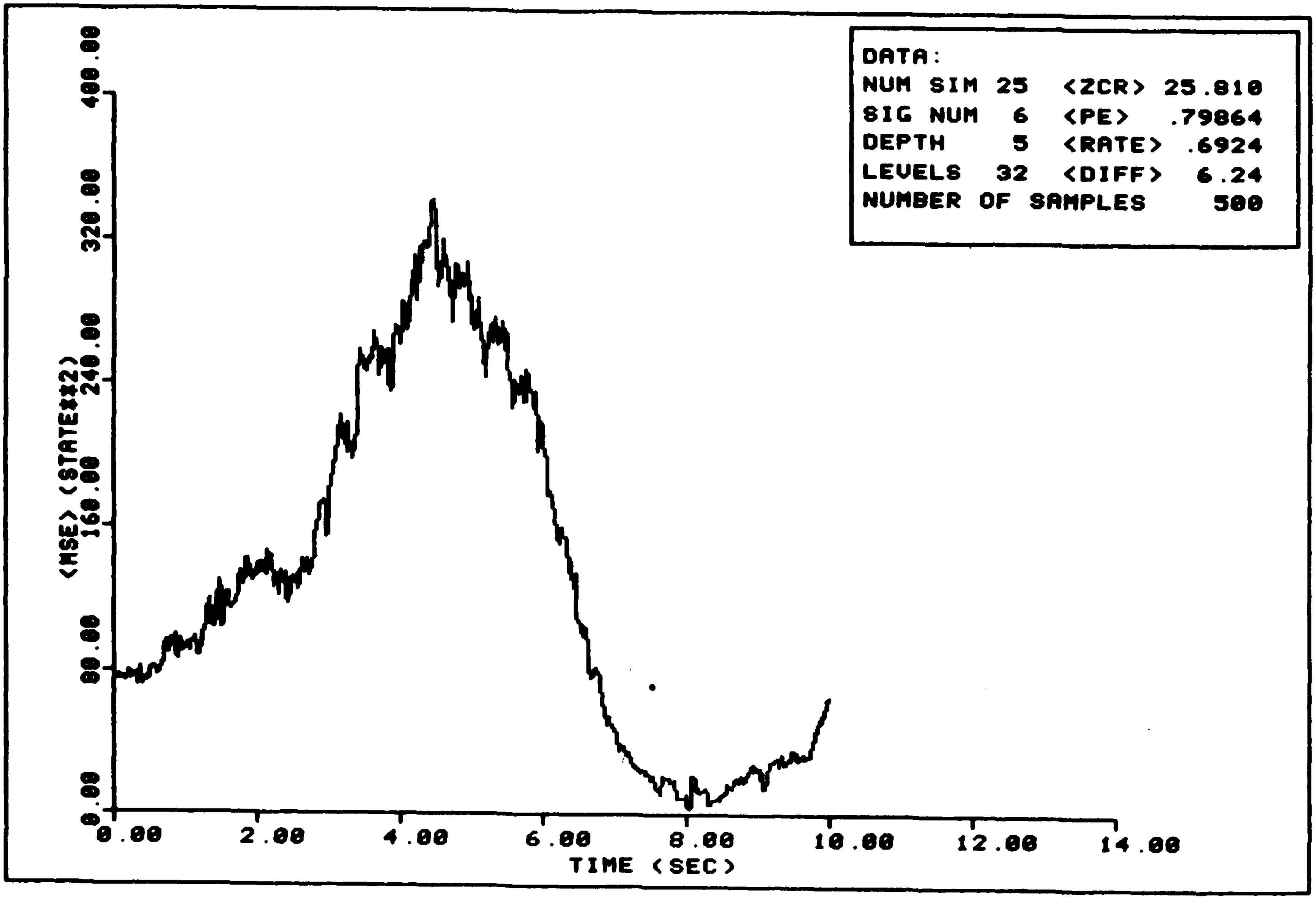


FIGURE 34. Ensemble Average MSE Performance

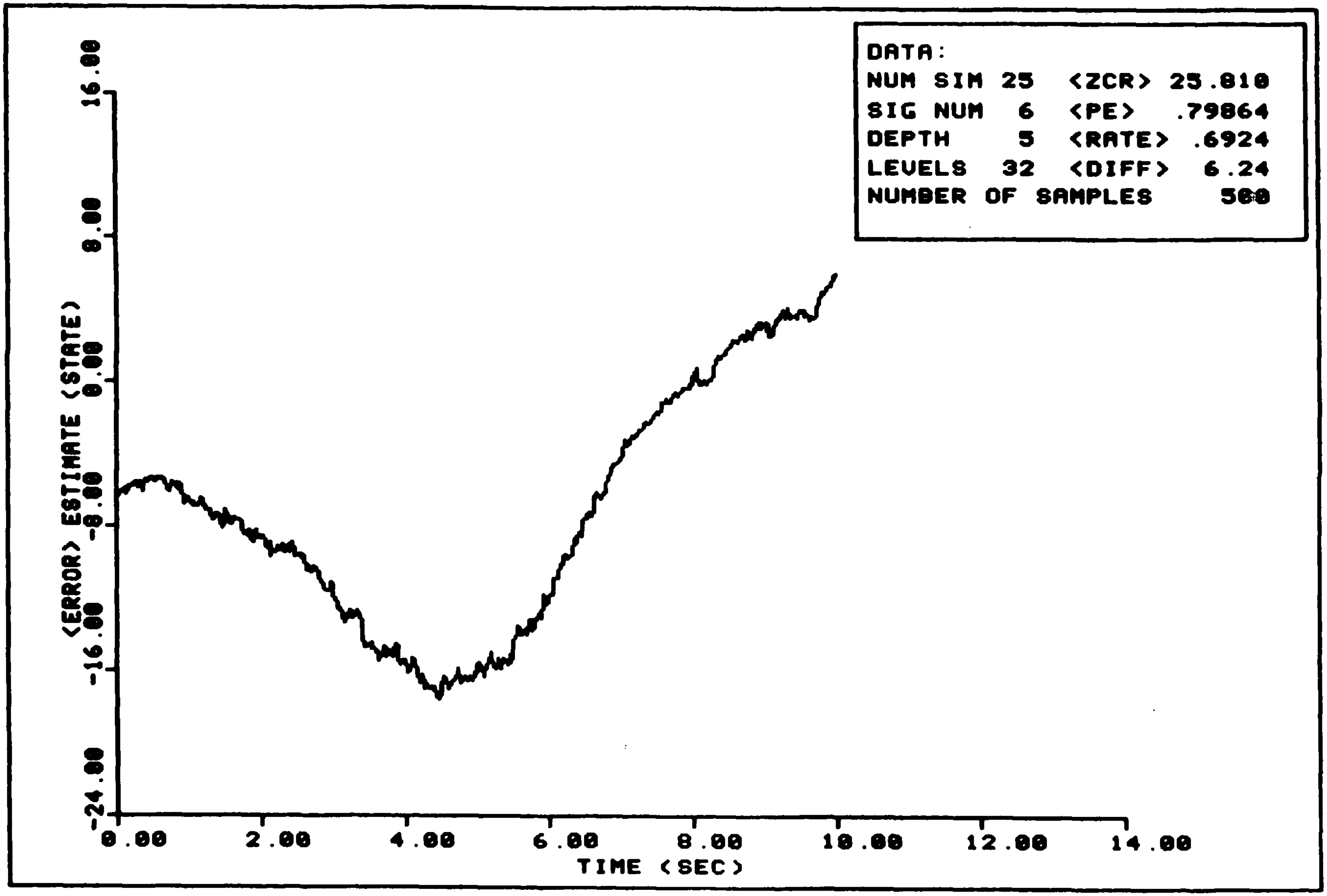


FIGURE 35. Ensemble Average State Error Performance

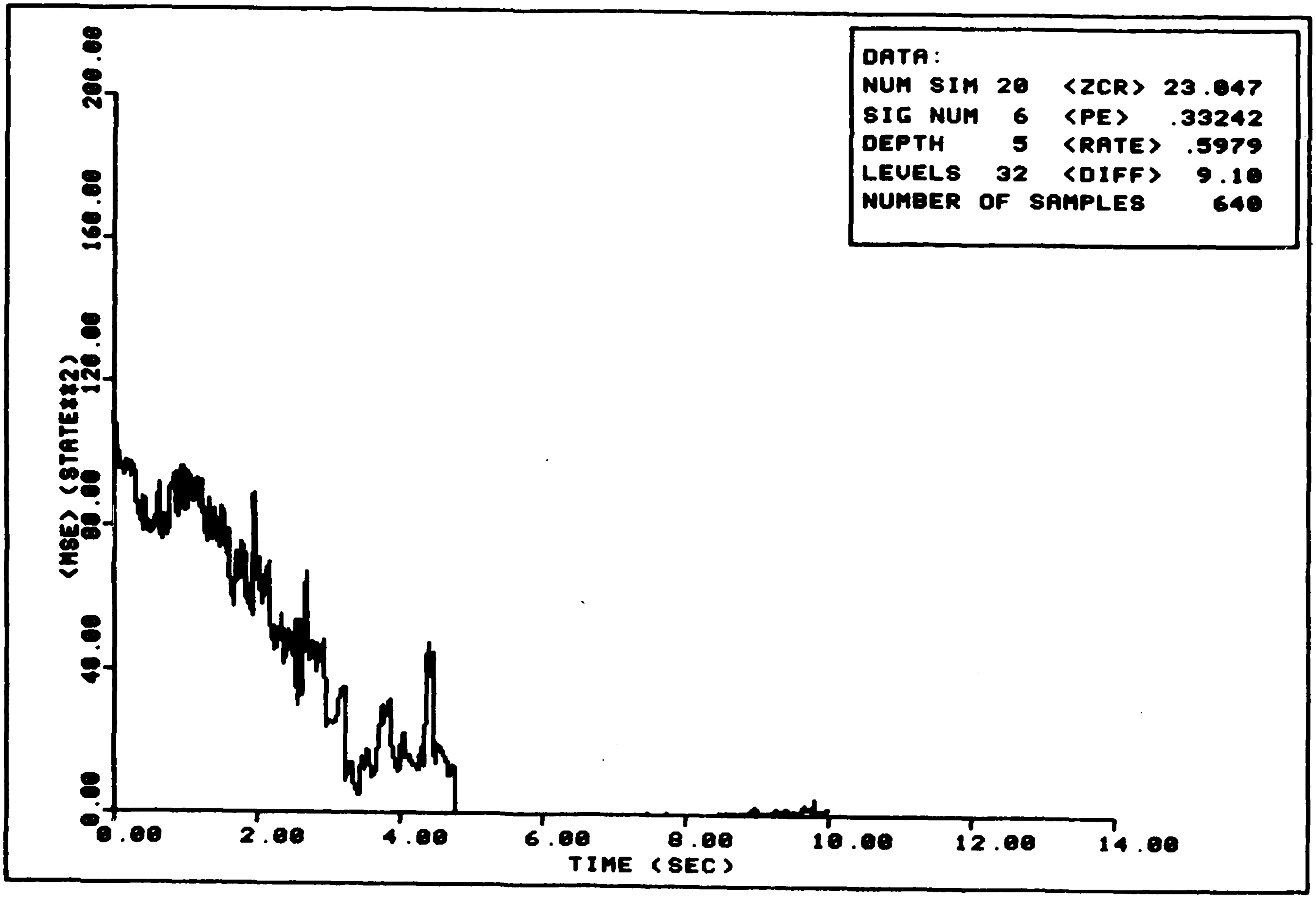


FIGURE 36. Ensemble Average MSE Performance

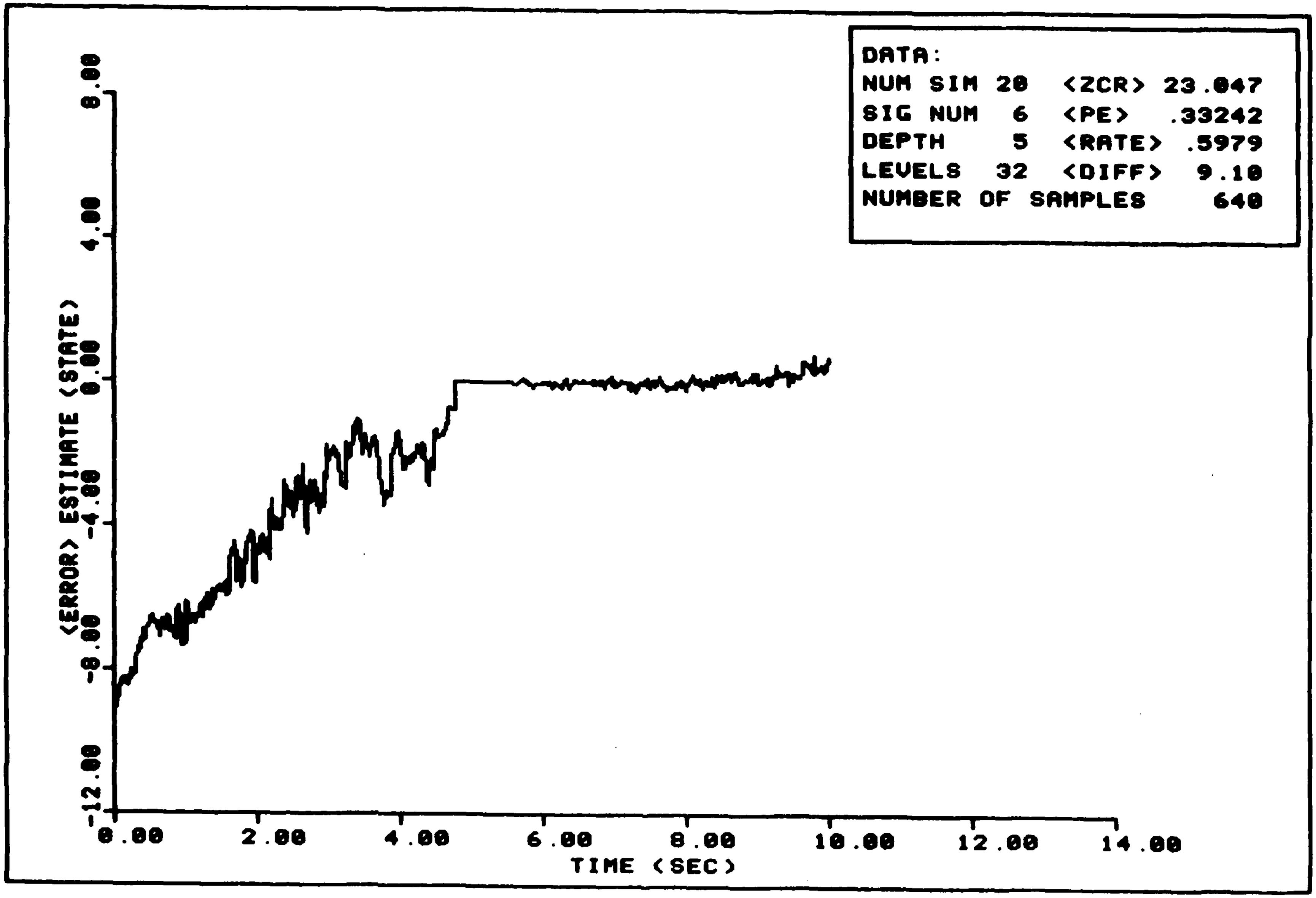


FIGURE 37. Ensemble Average State Error Performance

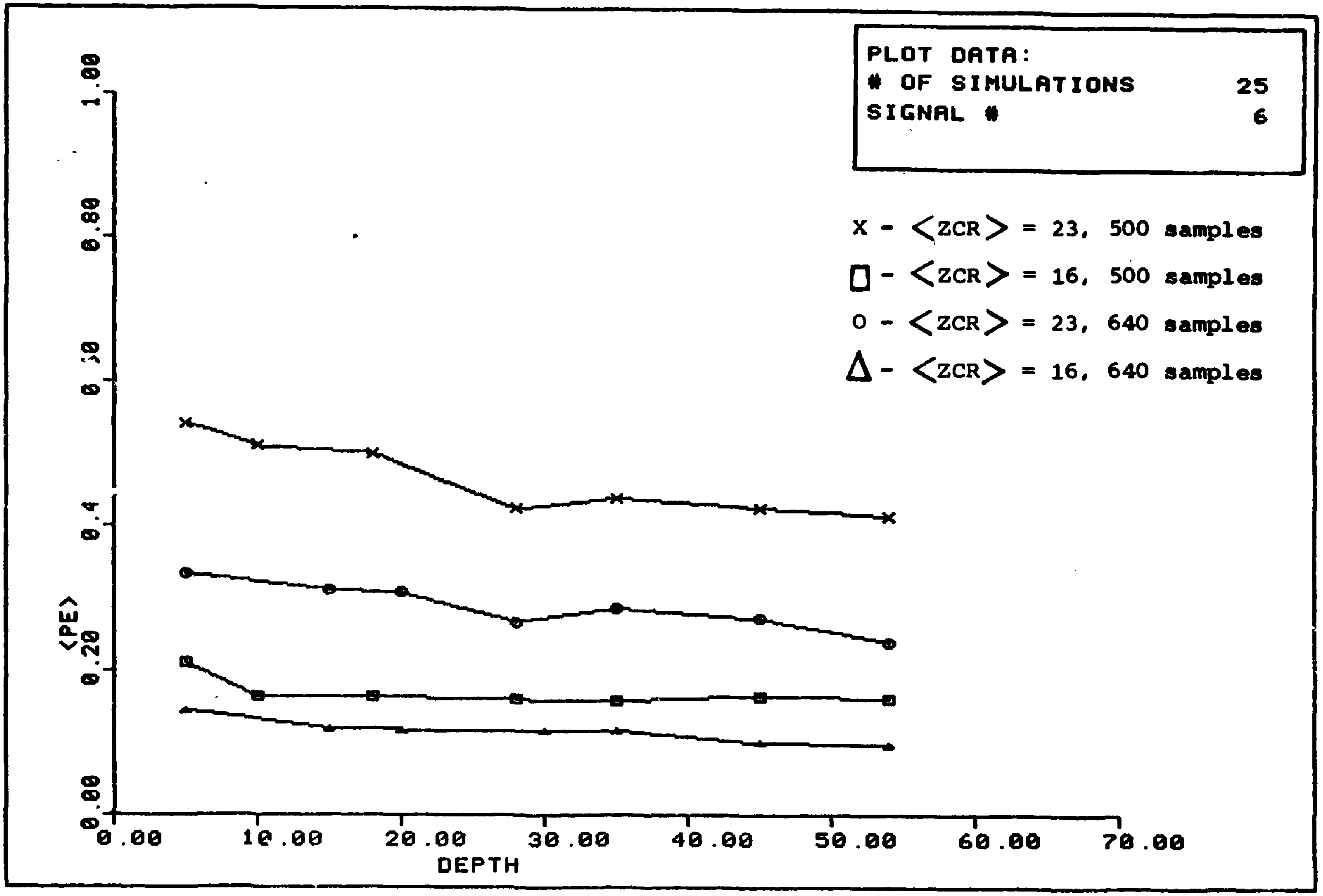


FIGURE 38. <PE> Versus Depth, Vary <ZCR> , Number of Samples

relationship in Figure 38. This figure shows that minimal gain in performance is realized with a decision depth $\gg 54$. This value corresponds to twice the minimum decision depth hypothesized earlier (i.e., $J_{min} = 27$ for $\xi_1 = 5$). Figure 38 graphically expresses the interrelationship between the sample rate and decision depth. This is similar to the relationship between decision depth and constraint length when the Viterbi algorithm is used for convolutional decoding (Ref 2). A decision depth beyond three or four constraint lengths produces negligible improvement in performance (three or four constraint lengths being the extent of the memory usable by the algorithm). A sufficient decision depth will be referred to as twice the minimum decision depth for the given f_S=64 S/S.

The ensemble average mean-square error and ensemble average state error for J_o=54 and f_S=64 S/S are shown in Figures 39 and 40. The net effect of increasing the sample rate and decision depth becomes more apparent by comparing Figures 39 and 40 to Figures 22 and 23. The average probability of error has been reduced from $\langle PE \rangle = .5432$ to $\langle PE \rangle = .2397$. The performance for $\langle ZCR \rangle = 23$ has been enhanced approximately to the level of $\langle PE \rangle$ for $\langle ZCR \rangle = 16$ (Figures 20 and 21). Sample estimations, showing the improvement in performance are provided in Figures 41, 42, and 43. The improvement is most noticeable when these figures are compared to Figures 30, 31 and 32; both sets are for $\langle ZCR \rangle = 23$.

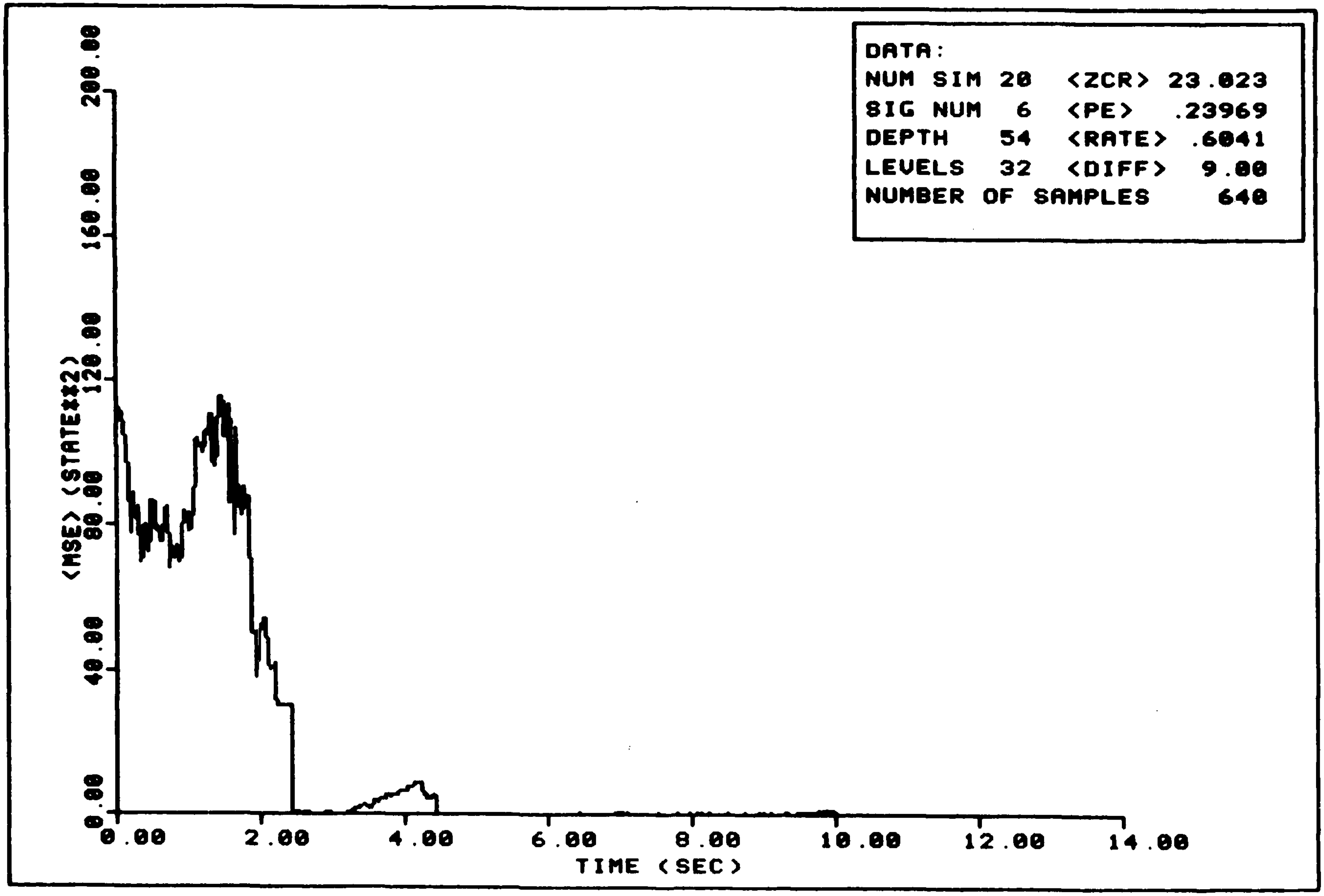


FIGURE 39. Ensemble Average MSE Performance

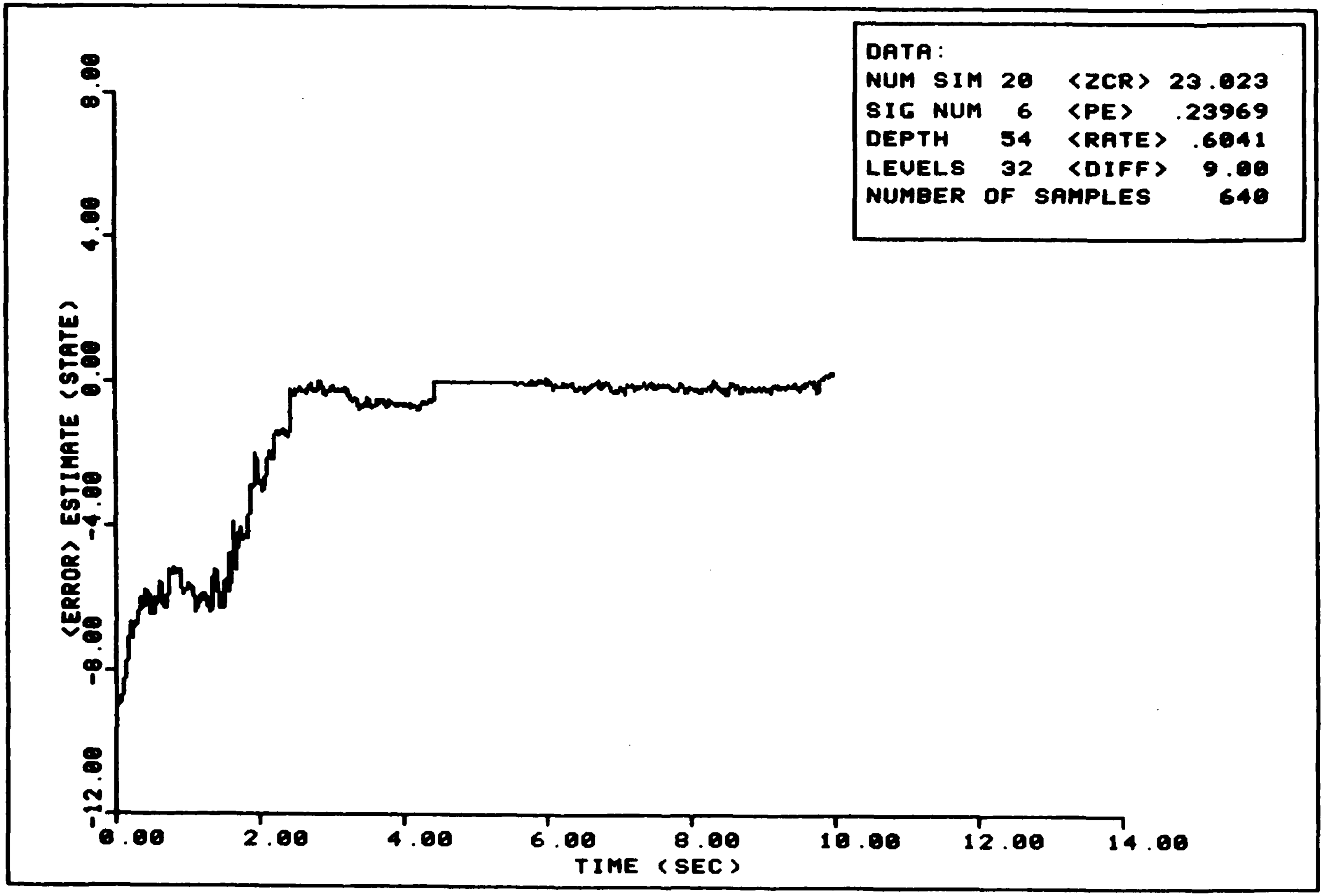


FIGURE 40. Ensemble Average State Error Performance

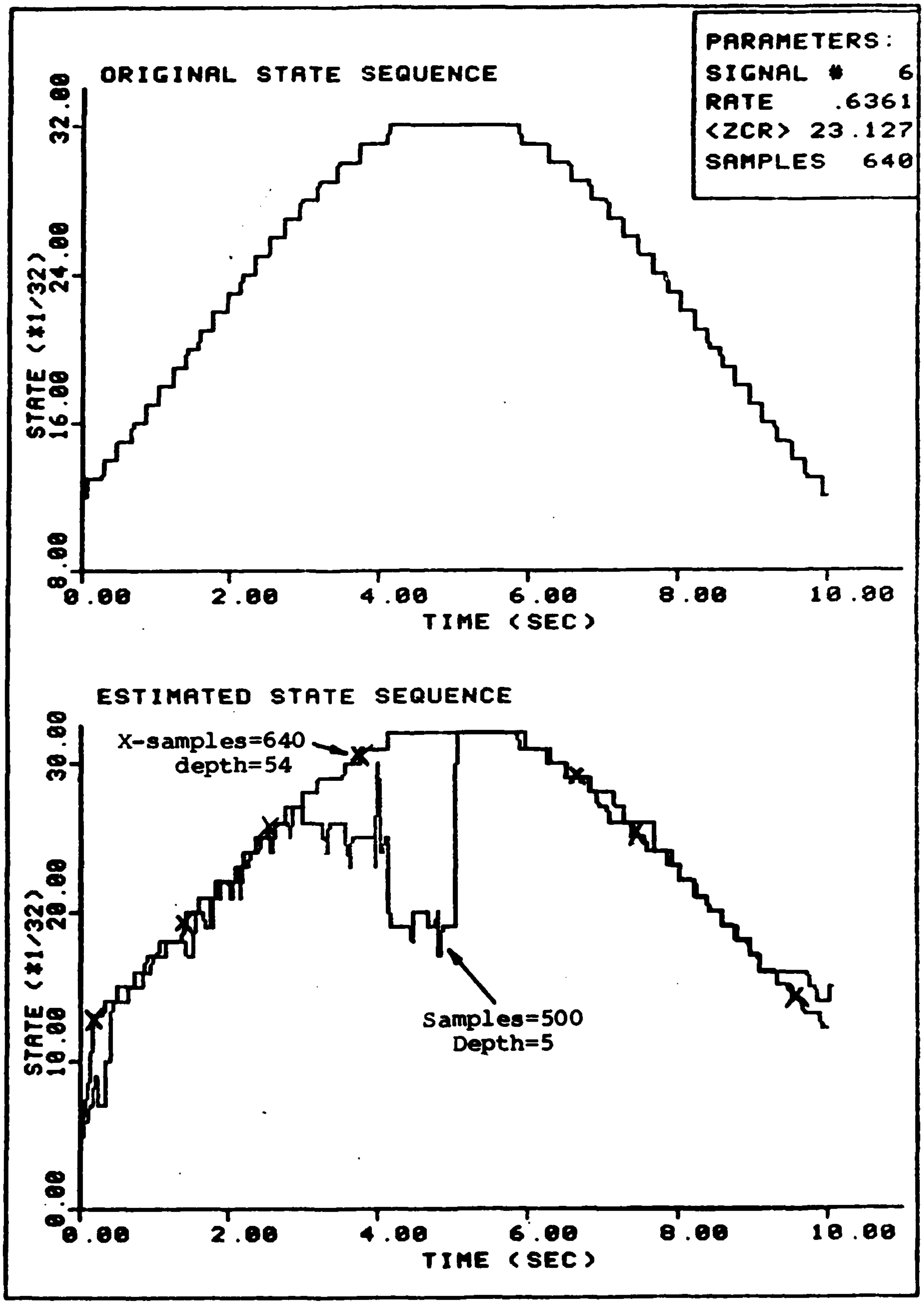


FIGURE 41. Sample Estimation - Performance Improvement

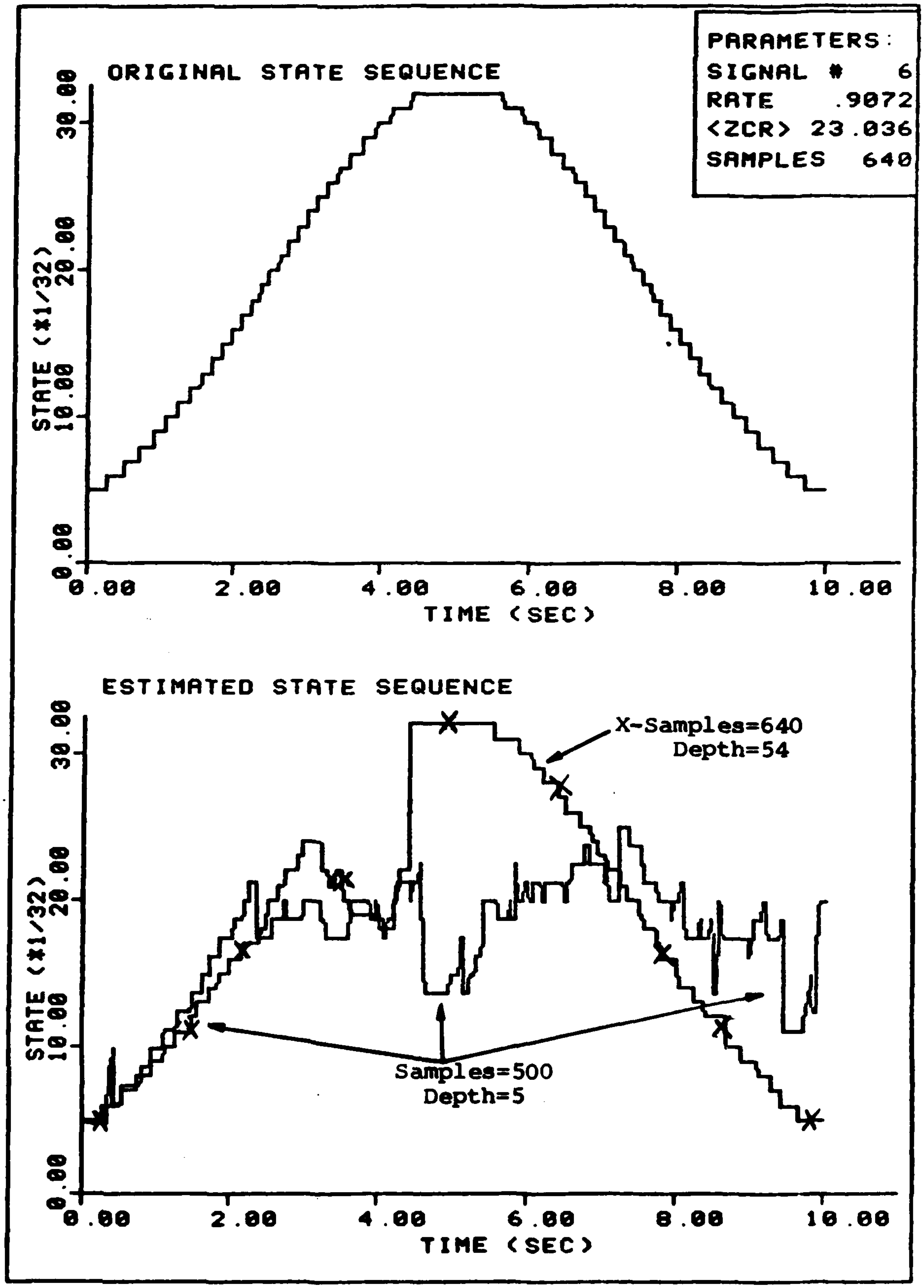


FIGURE 42. Sample Estimation - Performance Improvement

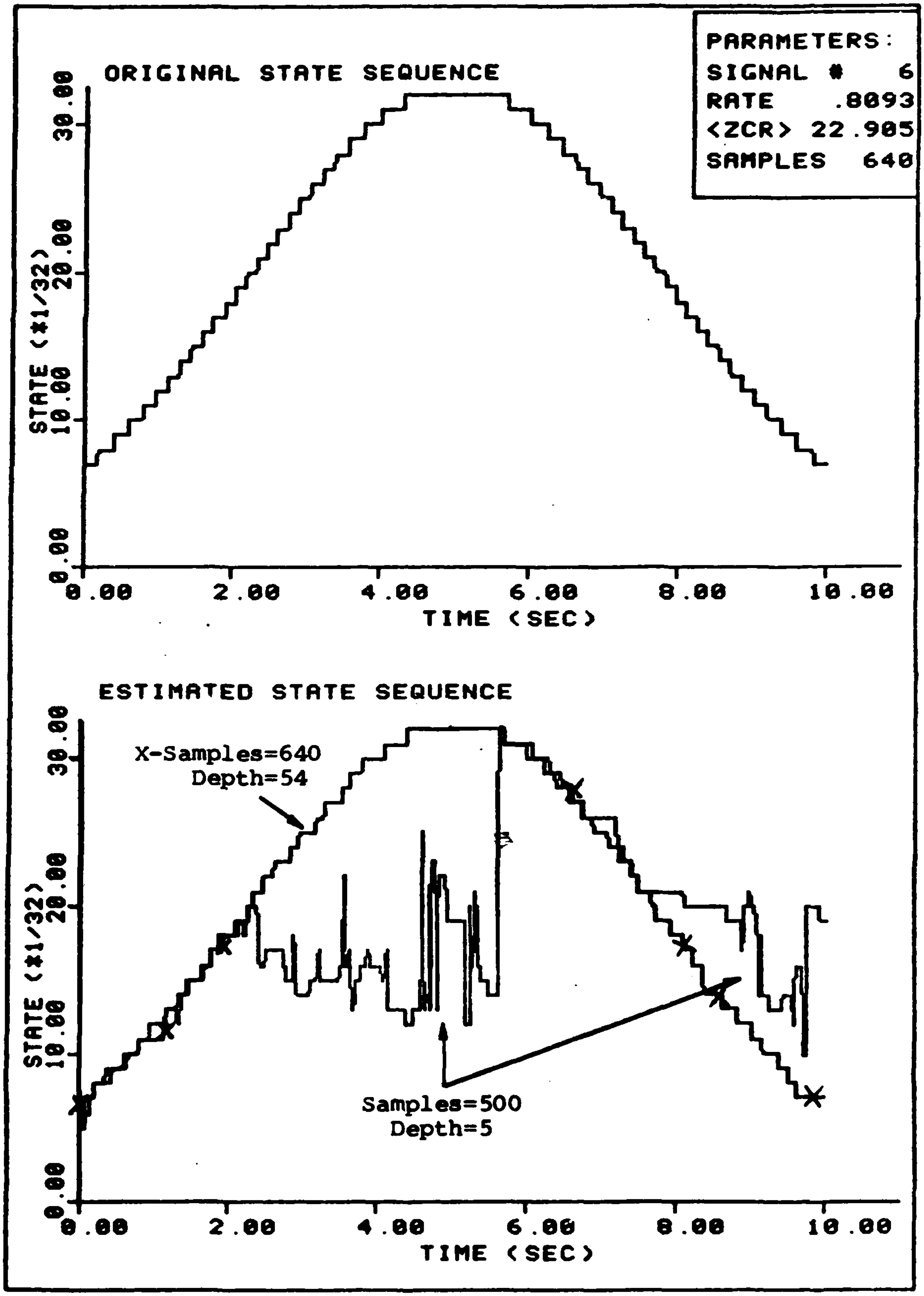


FIGURE 43. Sample Estimation - Performance Improvement

(4) Model Sensitivity. The assumed model is used to derive the priors for the shortest path solution. The sensitivity of the estimator to the chosen model was investigated for $\langle ZCR \rangle = 8$ and $\langle ZCR \rangle = 23$. For each test, a sample rate of 64 S/S and J_o=30 was used. To test the sensitivity of the algorithm, an estimate of signal #6 was attempted using equations (59) and (60) instead of equation (58). The results are shown in Figures 44-51. For both cases, the performance is worse for $\langle ZCR \rangle = 23$. It is interesting to note that for $\langle ZCR \rangle = 8.5$, the algorithm is quick to recognize the part of the signal sequence that resembles the process as described by equation (59) or (60). When equation (60) is used, the algorithm catches up to the sequence on the downward slope c the signal (resembles a pure death process) as in Figure 47. When equation (59) is used (pure birth process), the algorithm follows the sequence until the transition point and then begins to wander as in Figure 45. The estimator takes the entire observation period to catch up when $\langle ZCR \rangle = 23$ and equation (60) is used, as shown in Figure 53. Similarly, when $\langle ZCR \rangle = 23$ and equation (59) is used, the estimate is marginal on the upward slide of sequence and the wandering on the downward side is more pronounced, as shown in Figure 49.

The figures reinforce the expected sensitivity of the algorithm to the model. This model sensitivity should not be confused with mere sensitivity to source statistics (Ref

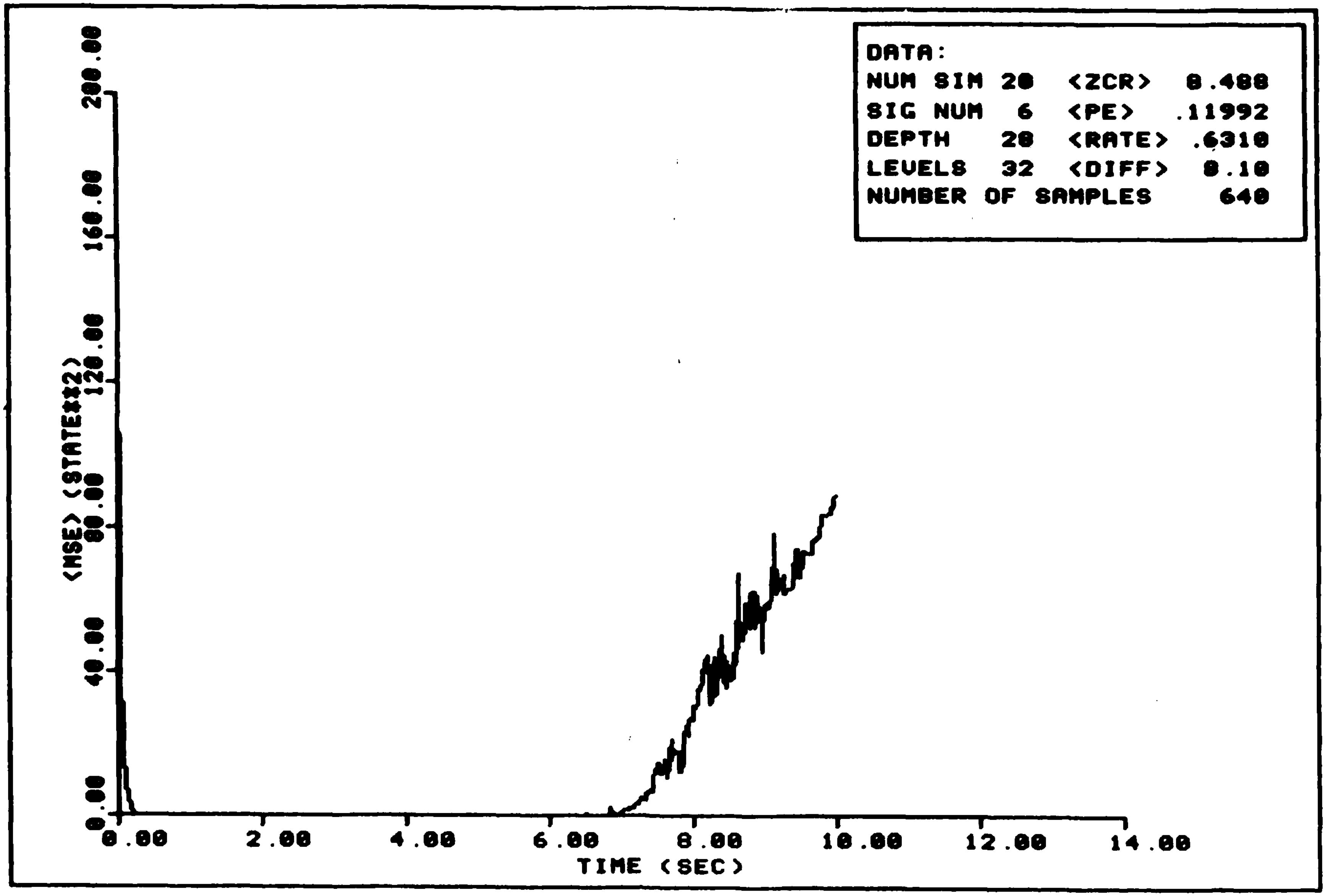


FIGURE 44. Model Sensitivity - Pure Birth Statistics

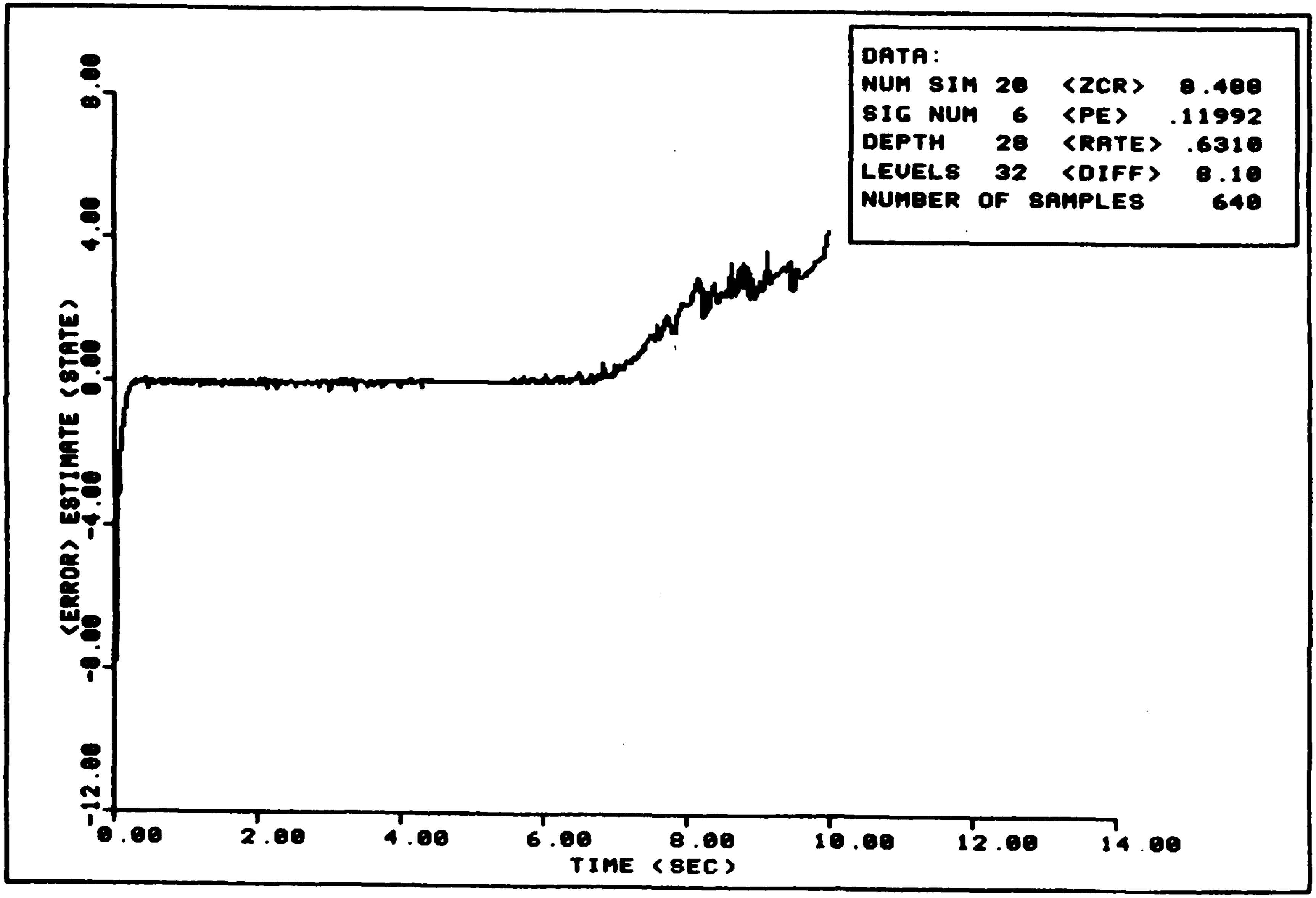


FIGURE 45. Model Sensitivity - Pure Birth Statistics

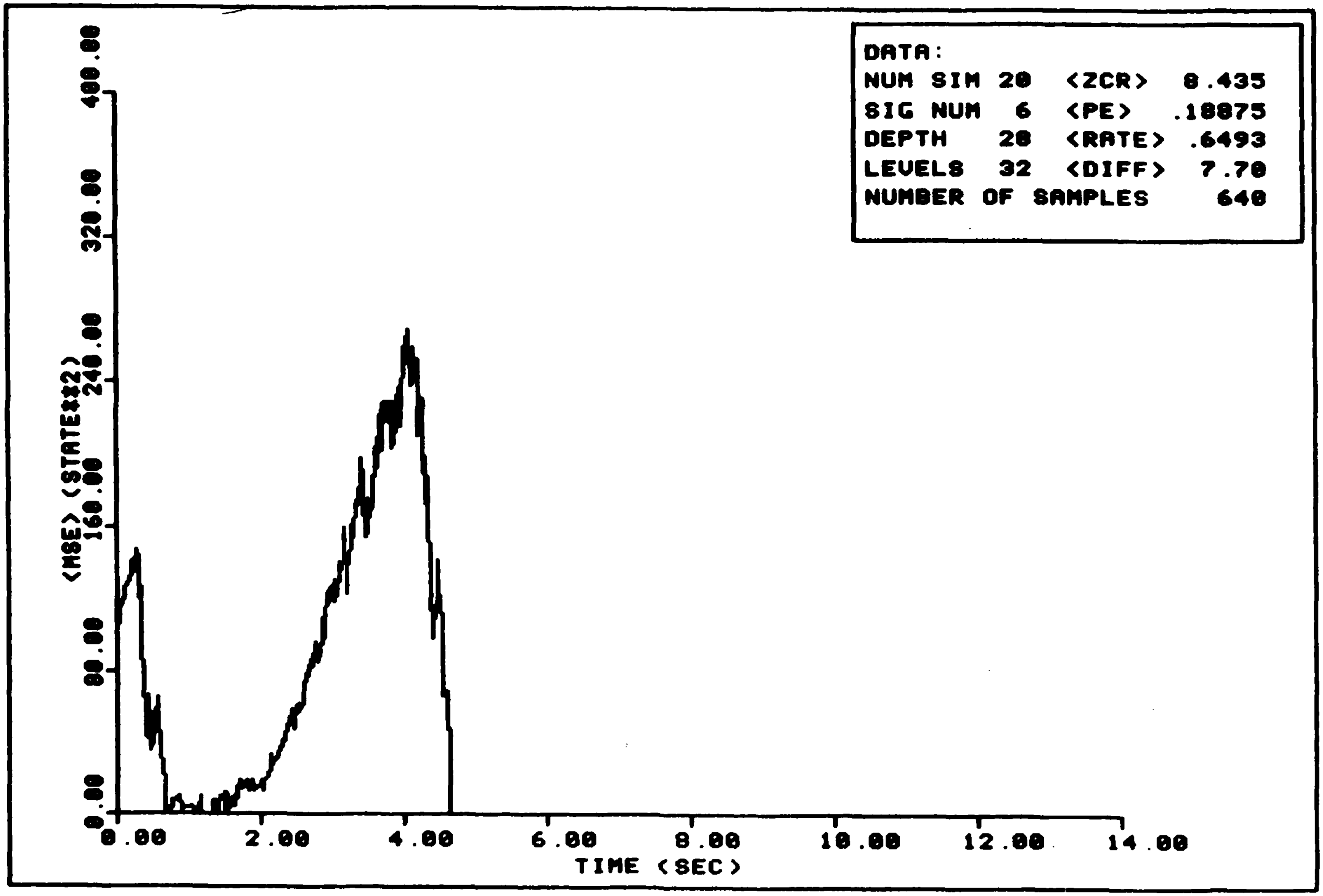


FIGURE 46. Model Sensitivity - Pure Death Statistics

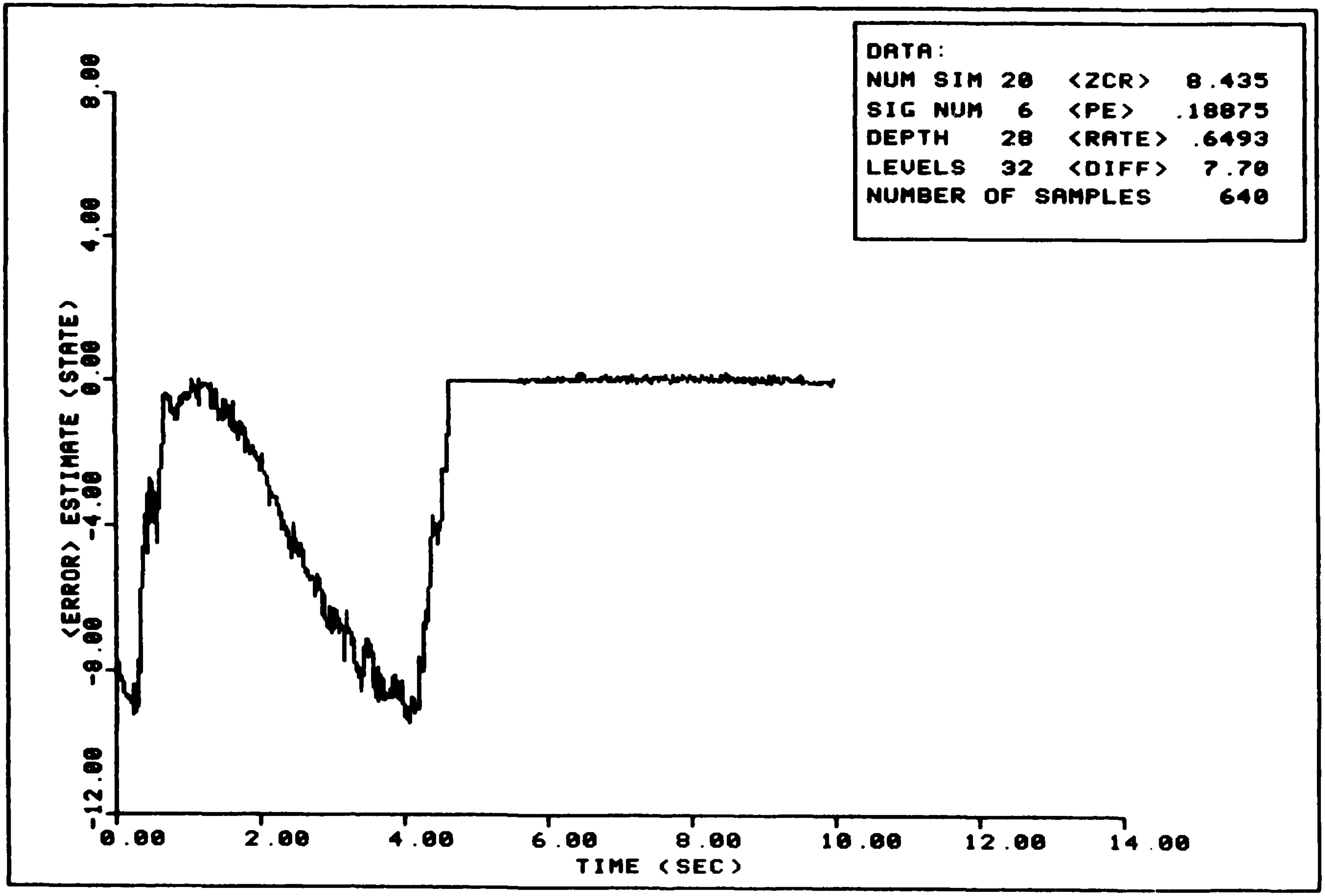


FIGURE 47. Model Sensitivity - Pure Death Statistics

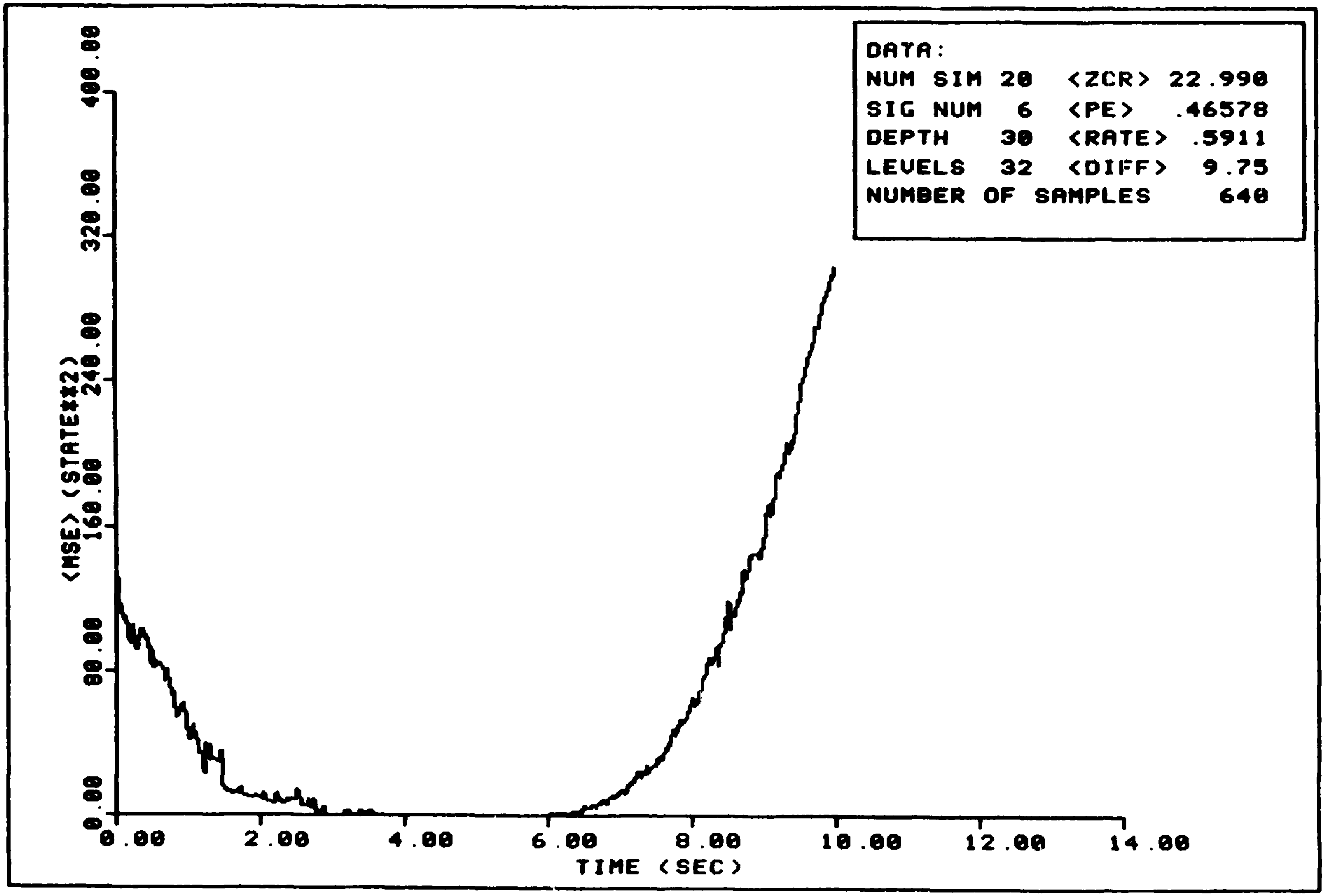


FIGURE 48. Model Sensitivity - Pure Birth Statistics

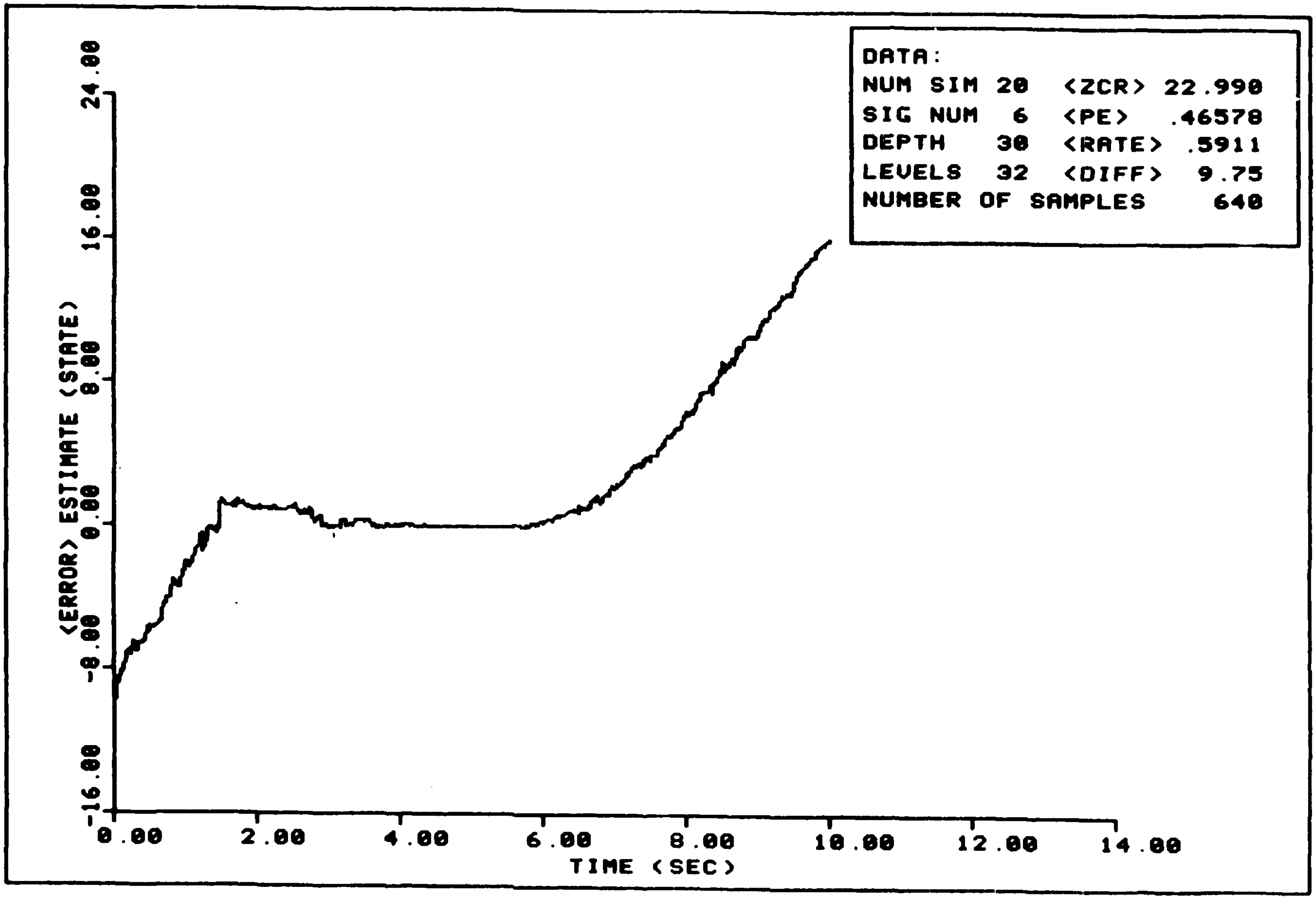


FIGURE 49. Model Sensitivity - Pure Birth Statistics

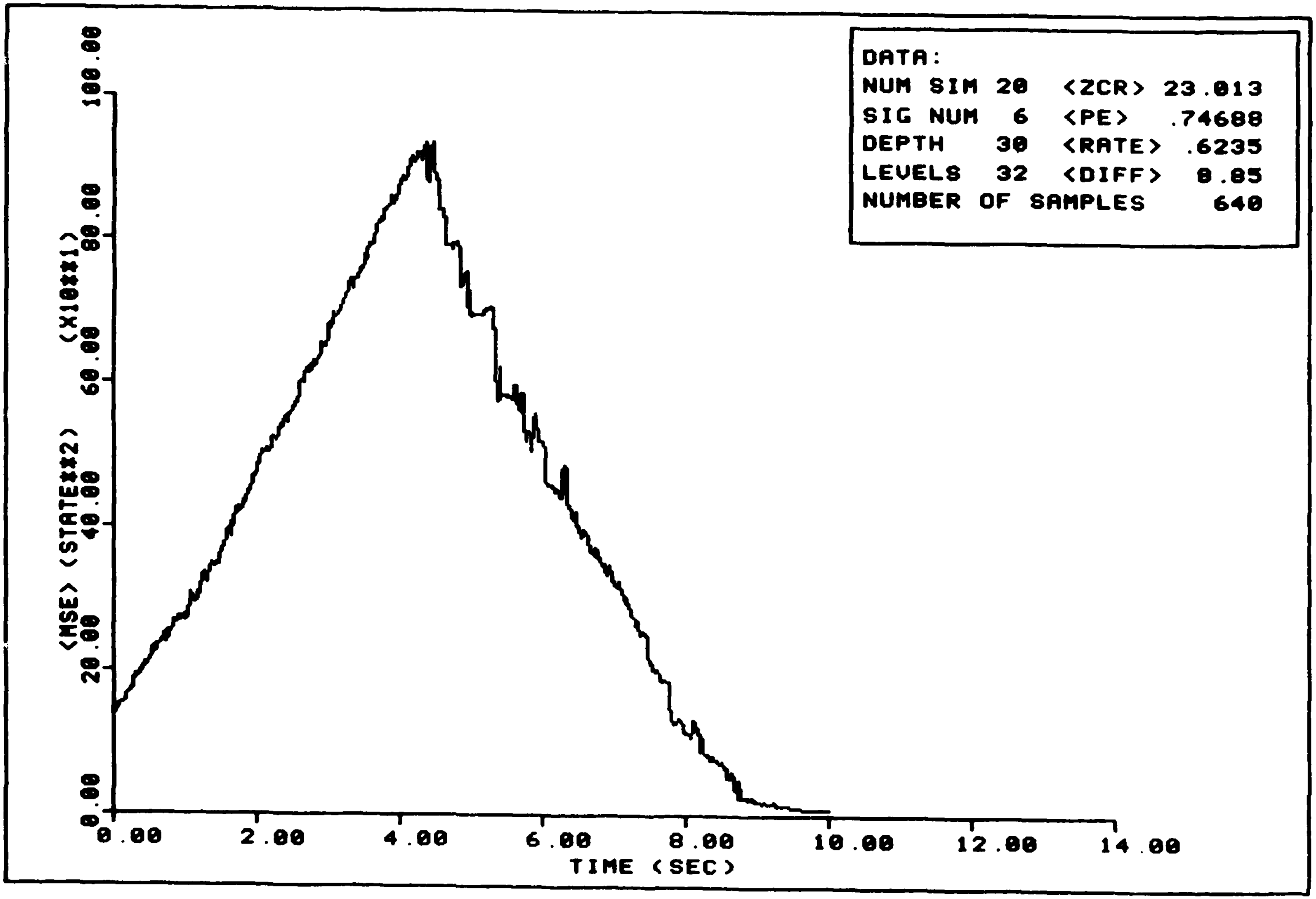


FIGURE 50. Model Sensitivity - Pure Death Statistics

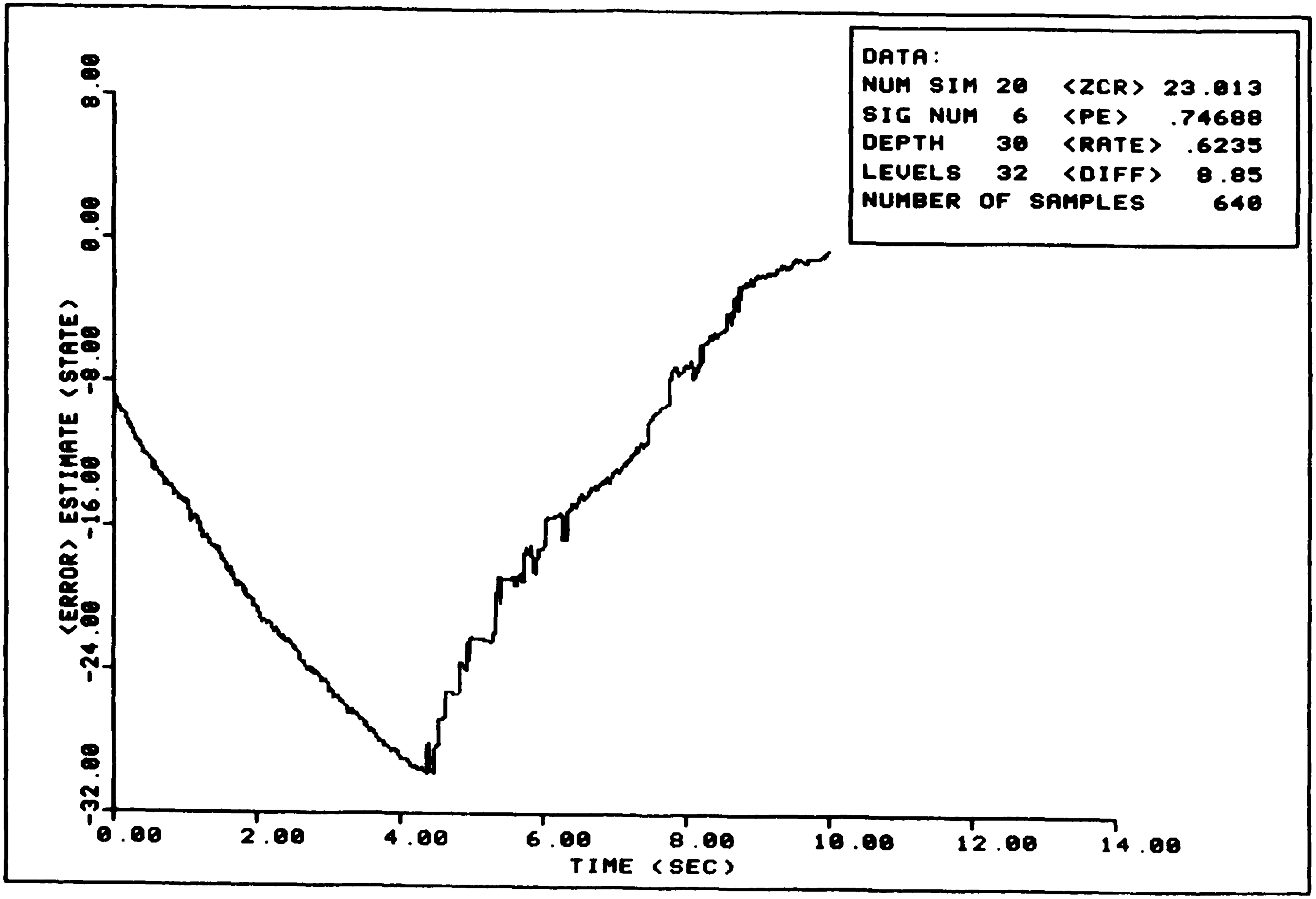


FIGURE 51. Model Sensitivity - Pure Death Statistics

16). Sequences modelled by equations (59) and (60) are no longer represented by a tri-diagonal matrix similar to equation (31), although the relationship of equation (29) is still used. The possible transitions have been reduced in terms of the transition probability values. Three possible transitions still exist, however, only two exist with any finite probability. Therefore, changing the model does more than change the source statistics. Ideally, if $\mu = 0$ in equation (59), and $\lambda = 0$ in equation (60), the entire trellis is modified as only two transitions are possible, and equation (31) becomes a bi-diagonal matrix.

(5) Rate Sensitivity. The sensitivity of the algorithm to the rate parameter is similar to the sensitivity to source statistics discussed by Ref 16. Using equation (58), a different rate parameter was used to generate the signal sequence and the transition probability. The same parameters were used as for the model sensitivity tests. Each of the two rates used were generated randomly and independently. The results are shown in Figures 52-57 for various $\langle ZCR \rangle$ values. No degradation in performance is observed when compared to performance under the conditions of equal rate parameters. The estimator still exhibits marginal performance for $\langle ZCR \rangle = 23$ but acceptable performance when the $\langle ZCR \rangle \leq 20$ (comparing performance when f_S=64 S/S and J_o=30). Figure 58 shows a comparison of performance based on increasing $\langle ZCR \rangle$. As expected, the algorithm is insensitive to the rate parameter used.

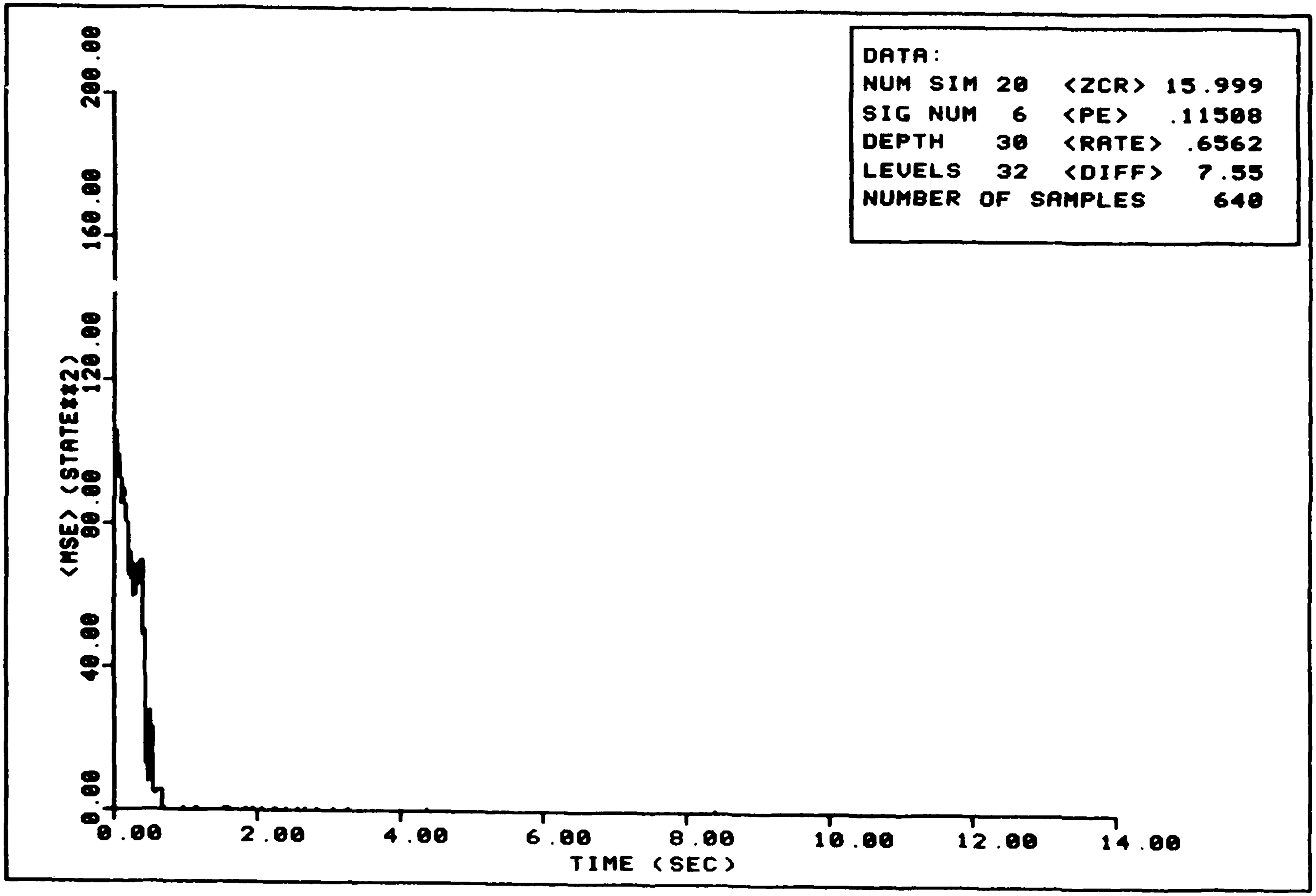


FIGURE 52. Ensemble Average MSE Performance - Random Rate Parameter

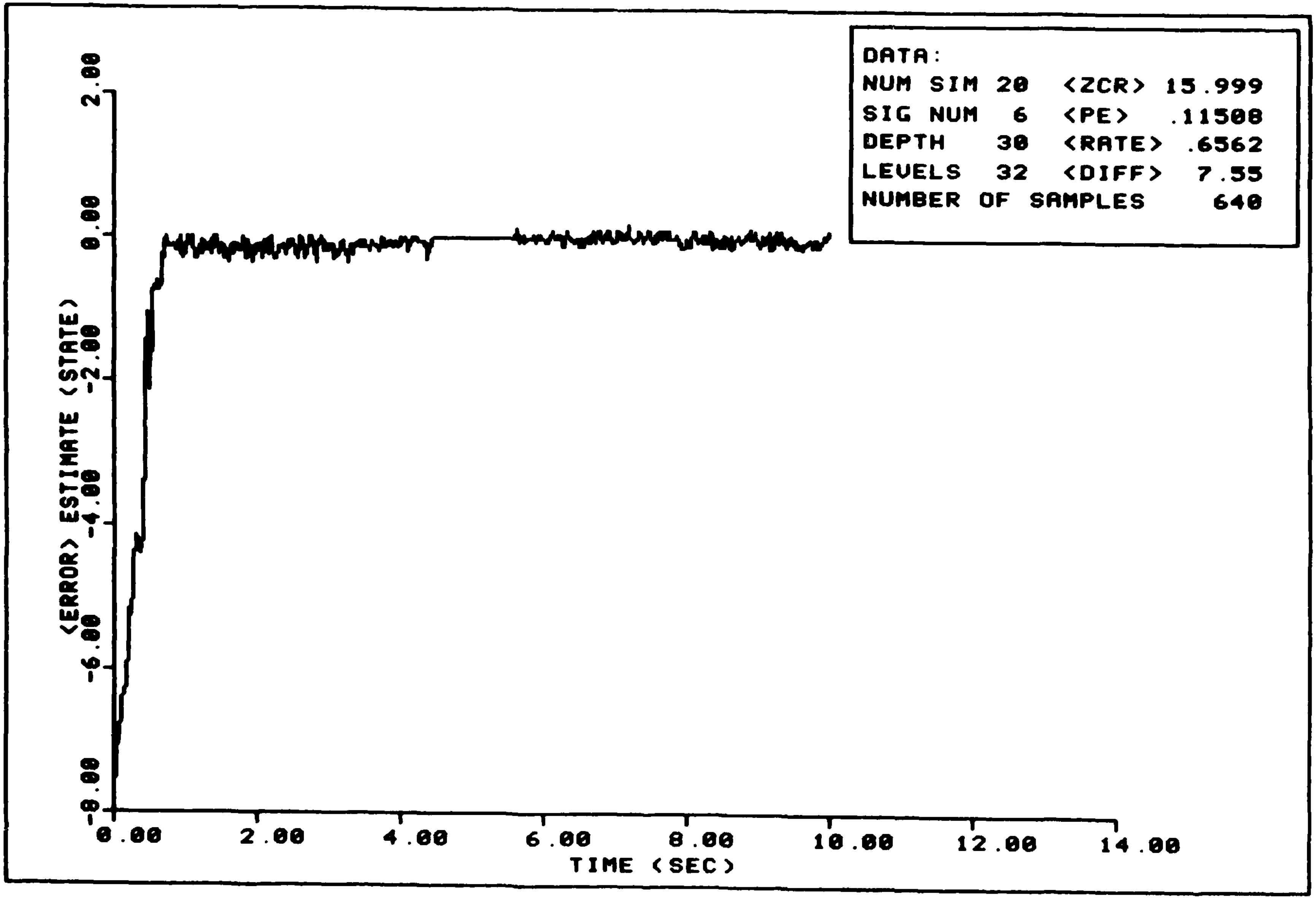


FIGURE 53. Ensemble Average State Error Performance - Random Rate Parameter

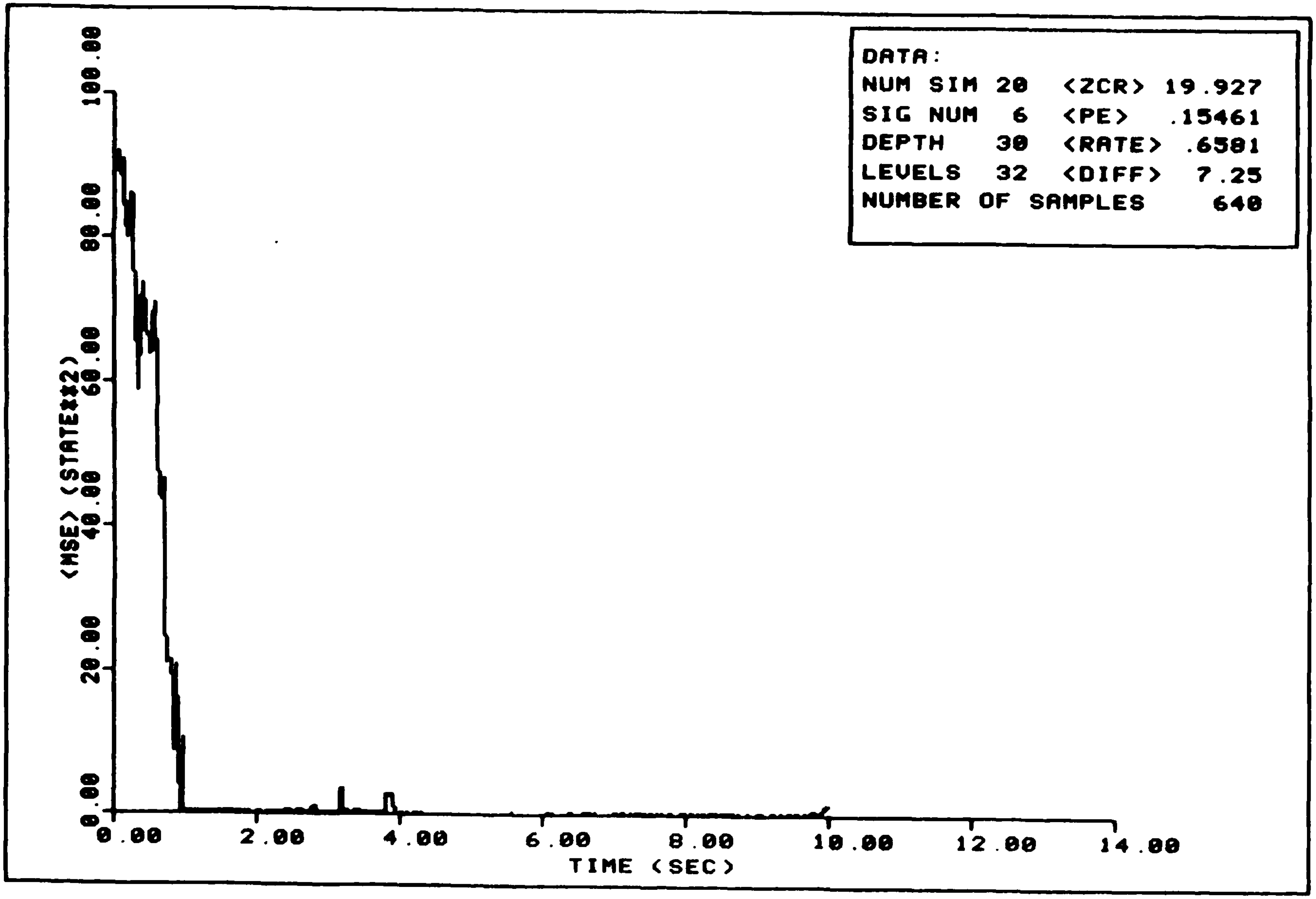


FIGURE 54. Ensemble Average MSE Performance - Random Rate Parameter

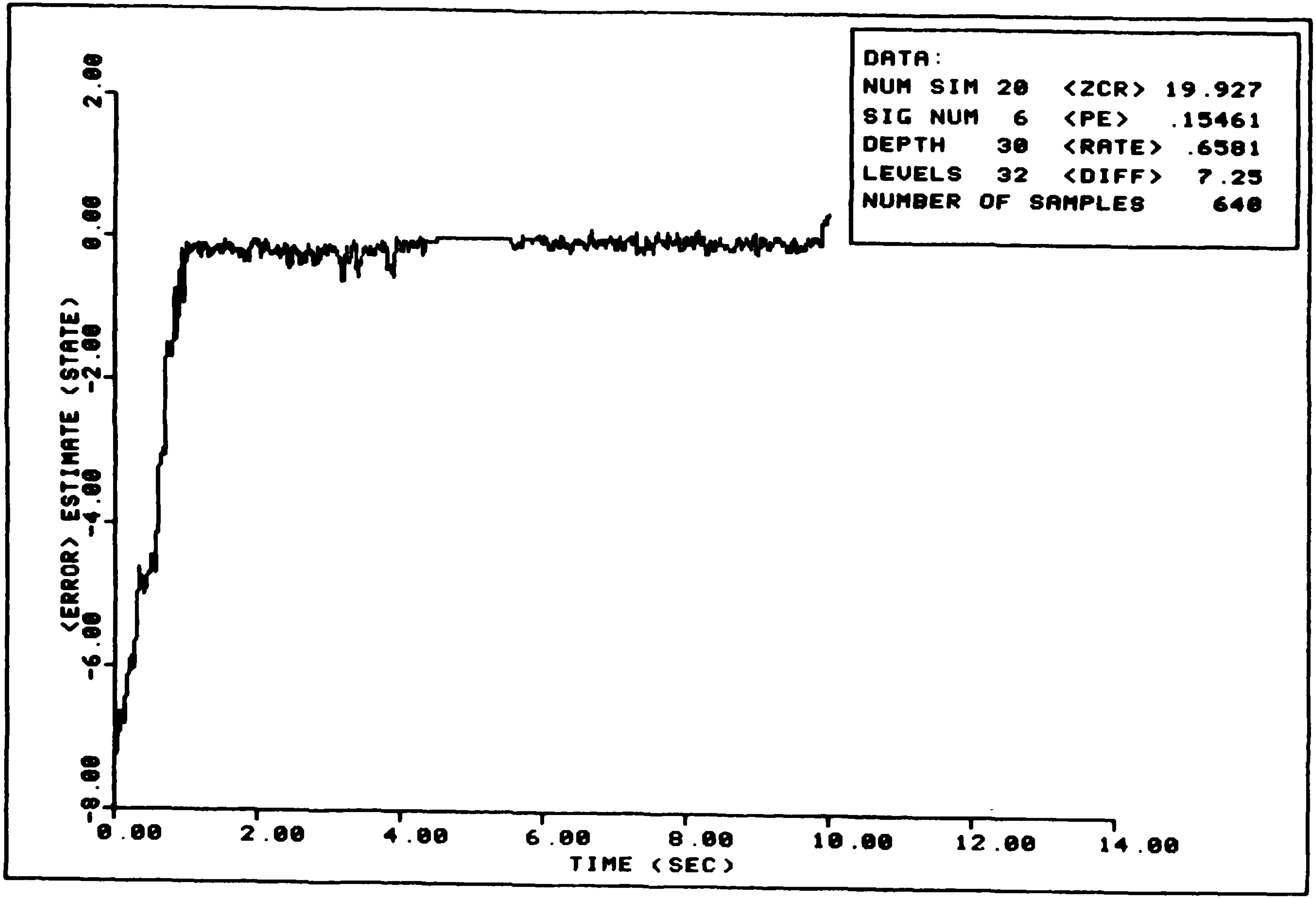


FIGURE 55. Ensemble Average State Error Performance - Random Rate Parameter

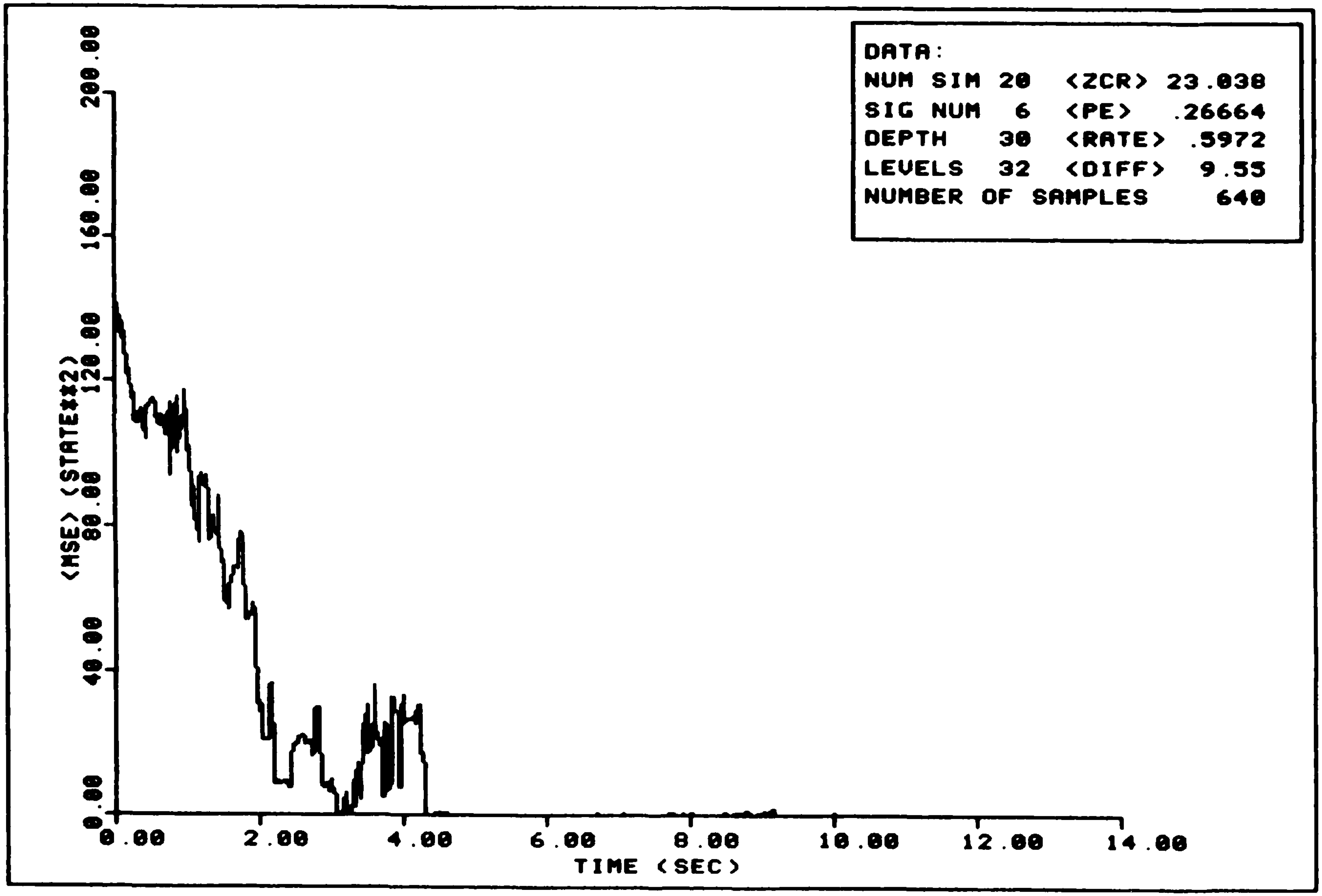


FIGURE 56. Ensemble Average MSE Performance - Random Rate Parameter

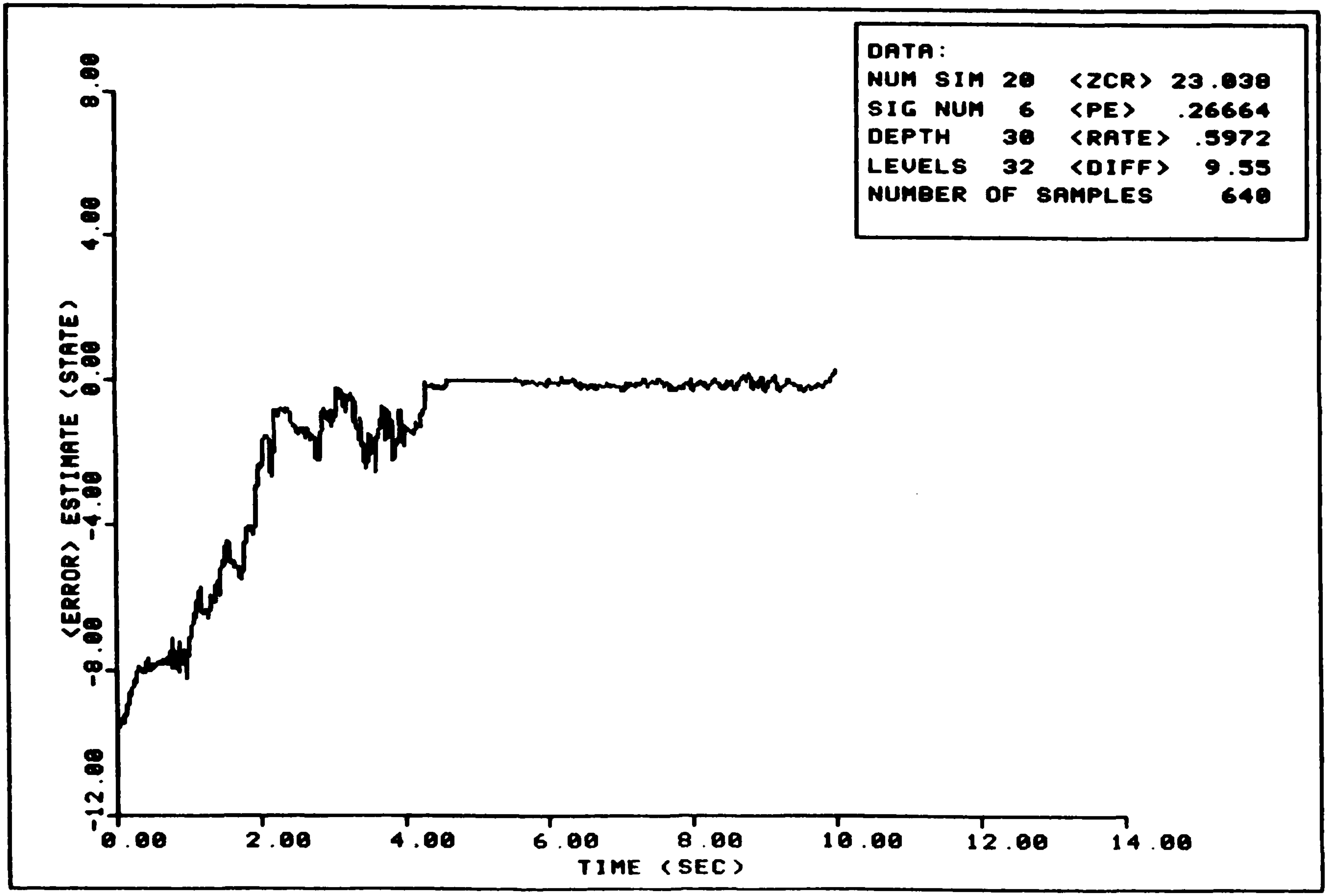


FIGURE 57. Ensemble Average State Error Performance - Random Rate Parameter

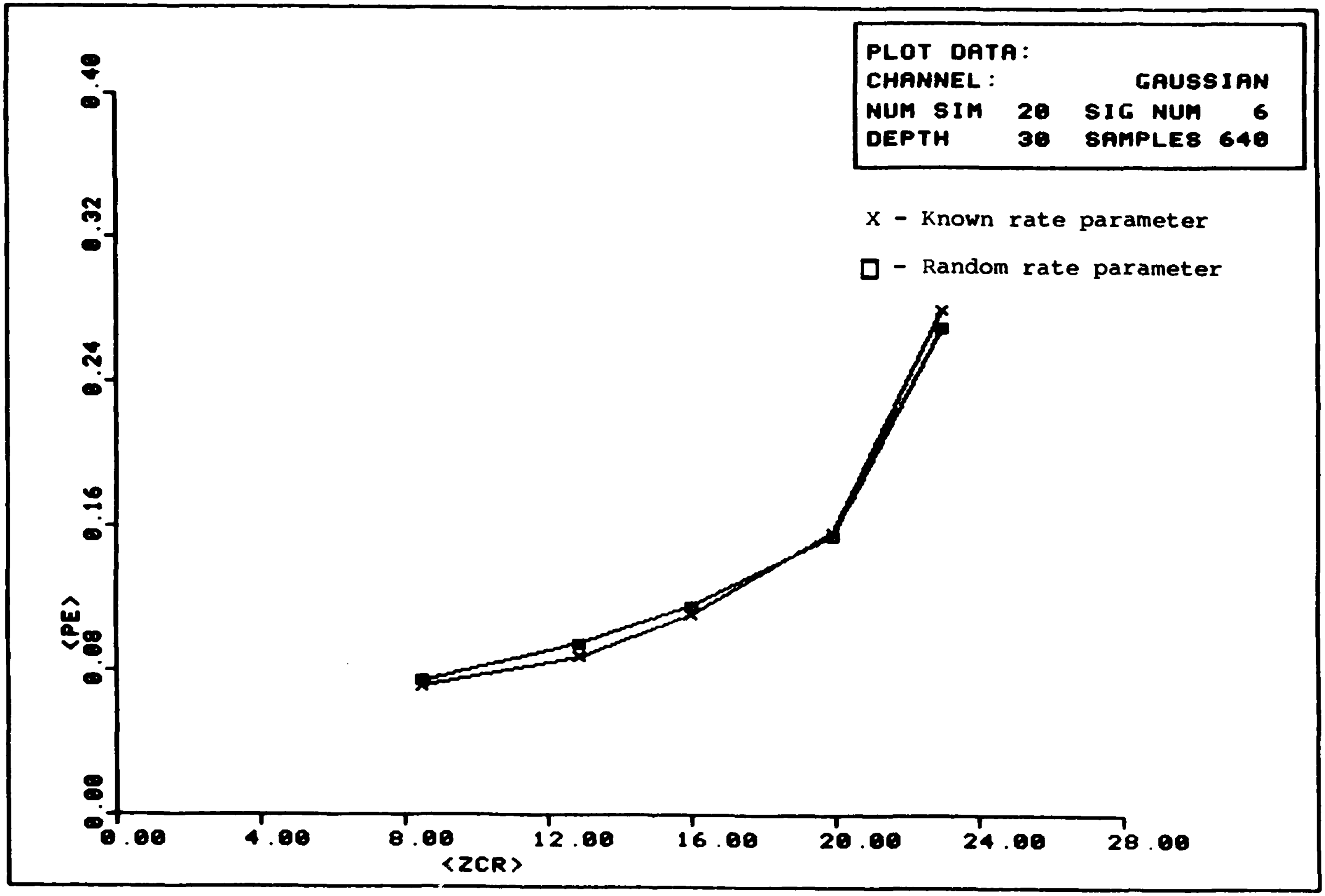


FIGURE 58. Rate Sensitivity Comparison

(6) Signal Set Performance. Estimations of the remaining signals (7, 8, 9, 10, and 3) were performed for various $\langle$ZCR$\rangle$ values. In all cases, performance varied little from that of signal #6. These simulations are provided in Appendix C. In general, the performance of the estimator for each signal lagged that of signal #6. Significantly, poor performance was observed for signal #10. This is based, for the most part, on its transition periods. Unlike the other signals (6, 7, 8, and 9), signal #10 exhibits a rapid transition from upward to downward slope. Most of the estimator error is spread over this transition region. Additionally, adequate reconstruction of the entire ensemble could not be achieved for as high a $\langle$ZCR$\rangle$ value. The sample estimates provided exhibit a cross section of tracking problems incurred for each signal.

(7) Comments. The data provided in this section reinforce the modelling of an adequately sampled signal as a discrete time, finite state, birth-death process. Expressing the signal sequence as such a first order Markov process allows, the Viterbi algorithm (dynamic programming) to be used to provide maximum a posteriori estimate of the signal. In this sense, the estimator may be assumed optimum and unbiased.

The data presented, based on the birth-death process model, indicate that relationships exist between model and algorithm parameters, similar to those established for convolutional encoding/decoding utilization of the Viterbi algorithm.

The results indicate an ability to reduce the multipoint data with a high degree of confidence in the reduced data. The data presented indicate a limit to the ability of the estimator to provide single point data with a high degree of confidence, based on the assumed model and hypothesized parameter relationships. An exhaustive study, based on parameter variation, would be expected to provide more accurate parameter relationships.

The performance of the estimator for the entire message set is summed up in Figure 59. For each estimation, a sample rate of 64 S/S was used and a decision depth, $J_o = (2*J_{min})$ was used. Performance for the entire set is consistent in that the "break point" is for a range of $\langle ZCR \rangle$ greater than 21 and less than or equal to 23.

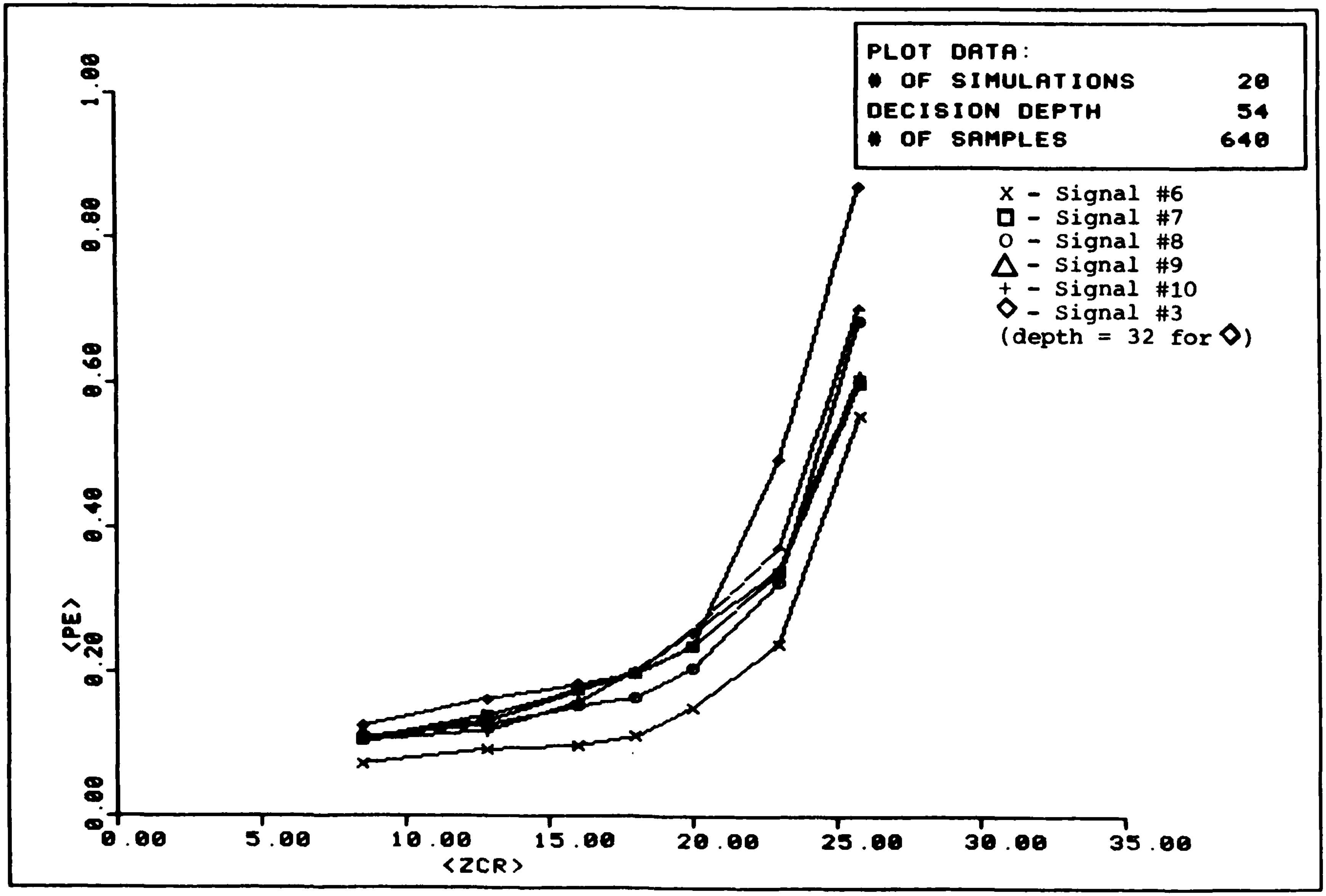


FIGURE 59. Message Performance Comparison

IV. CONCLUSIONS AND RECOMMENDATIONS:

This report has dealt with the problem of PPM, multipoint data reduction via amplitude/pulse-position sequence estimation in the presence of white Gaussian noise. A specific receiver structure and a unique signal model based on this structure, has been presented. The performance of a MAP estimator, based on a single Markov signal model, has been presented for a variety of messages signals. The performance of the estimator suggests that the "bit lock-out" technique may be a viable alternative to the frame lock-out technique for dealing with anamolous errors in PPM. As with any investigation or study, many new questions are asked. These questions become the impetus for further study.

CONCLUSIONS:

The primary objective of this study was to provide an automated technique for reducing the given multipoint PPM data to single point data with a high degree of confidence that the edited data accurately reconstructs the original message signal. The primary concern was a reconstruction that preserved the shape of the message rather than provide a point-to-point mapping of original to estimated messages.

The pulse-position sequence Viterbi algorithm estimator, with 32 pulse position quantization levels, has been shown to provide single point data that provide a reconstructed message with variable, but predictable, levels of confidence. Since the probability of at least one anamolous pulse occurring per frame = 1.0, the performance of the estimator has been expressed in terms of the total number of threshold crossings per frame (signal plus anamolies). The primary performance indicator was the average probability of estimator error, $\langle PE \rangle$, defined as the number of error estimates divided by the total number of estimates.

It is impossible to compare the bit lock-out estimator performance with other estimators. Instead, a relative performance rating, with respect to the objectives stated earlier, must be made. The estimator does provide an accurate reconstruction of the original message, based upon the assumed message set, and the general assumptions made regarding the behavior of the message. As expected, the ability of the estimator to provide a reconstruction with a high level of confidence deteriorates as the number of threshold crossings per frame increases. This was shown to be true for each of the six sample messages.

A relationship between the signal model parameters and the Viterbi algorithm parameters heretofore did not exist. An attempt to establish such a relationship was made in this study. In particular, a method for determining a minimum sampling rate in order to provide sufficient memory for the algorithm to utilize was hypothesized. Also, a minimum and sufficient decision depth was hypothesized based on the trellis representation of the Markov process model.

The hypothesized minimum samplе rate was shown graphically to far exceed the minimum rate needed to provide the ±1 limitation in pulse position from sample-to-sample for each of the sample messages. The performance indicators imply that the sample rate may be determined by equation (61) where K=2 provides the minimum sample rate (for the range of rate parameters used) that provides the memory needed by the algorithm. The decision depth is related to the trellis defining each sequence estimation. In particular, the minimum depth, as well as a "sufficient" depth was shown to be related to the initial starting state used to define the trellis. The decision depth is related to the maturation time (or depth) of the trellis which is based on the assumed initial starting state.

These hypothesized relationships were used to arrive at the limiting point of algorithm performance. It was shown that acceptable performance was possible when the average number of threshold crossing, $\langle ZCR \rangle$, was less than or equal to 23. Parameter variation, both in sample rate and decision depth, provided an improvement in performance as expressed in terms of $\langle PE \rangle$, for $\langle ZCR \rangle \leq 26$.

The estimator was shown to suffer from two types of errors while still providing a reasonably accurate estimate. These two being transient error and transition error. When the estimator fails to provide a reasonable facsimile of the sample message, these errors merge and become indistinguishable. The transient error and transition error are insignificant for $\langle ZCR \rangle \leq 16$. However, for $\langle ZCR \rangle \geq 16$, these errors become more pronounced. The effect of these errors may be reduced by:

(1) Providing more accurate initial starting state information to minimize the transient error.

(2) Increasing the sample rate (i.e., K value in (61) and the decision depth.

It was demonstrated that increasing the decision depth arbitrarily does little to improve performance, for a given sample rate.

RECOMMENDATIONS FOR FURTHER STUDY:

The following areas are recommended for further study:

(1) The model chosen to describe the signal sequence places restrictions on the type/behavior of the message. It may be possible to develop a Markov model to describe each sample message. In this way, the observation vector could be processed throtgh each model in parallel and the model sensitivity observed in this study used to provide the best estimate.

(2) It may be possible to develop a Markov model that does not require a sample rate that meets the adjoining state limitation. It is recommended that investigating Markov models that allow pulse position changes of ±2 or

even ±3 pulse positions from sample-to-sample. Such a model may allow for more complex (i.e., more rapidly varying signals, or those signals with a modulated amplitude like that depicted in Figure 4).

(3) It is recommended that a positive relationship be developed to tie the conventional utilization of the Viterbi algorithm to other MAP estimation problem applications. In particular, a relationship is necessary to define a minimum sample rate to provide the memory synonymous with that provided by a convolutional encoder. Also, an expression defining a minimum as well as a sufficient decision depth must be developed. This decision depth is synonymous with the "3 to 5 constrant lengths" used as the sufficient decision depth measure for convolutional decoding; however, the term constraint length has no meaning when the Viterbi algorithm is used to estimate a general Markov process (amplitude or phase estimator).

(4) In order to refine the proposed model or develop similar Markov models, a more precise relationship must be established between the rate parameter as it applies in determining the transition probabilities and as it applies to the message waveform itself. Once these relationships are established, an upper bound on performance may be established, such as one based on the shortest error path within a given trellis. Definitive statements regarding model suitability

and algorithm performance may be made once bounds on performance are established.

(5) It is recommended that different channel models be applied for this problem. The Viterbi algorithm assumes the channel is memoryless (white noise). This study utilized a Gaussian channel model. A nonzero mean probability density function, such as Poisson, with low count rates, may provide a more accurate description of the channel effects in producing numerous anamolous pulse errors. Once again, all parameters (signal model, channel model, algorithm) must be tied together to provide a measure on the estimator's performance.

(6) Lastly, it is recommended to investigate the effect of increasing the number of bit intervals per frame. An arbitrarily large number of bit intervals per frame may actually degrade performance. Since each bit interval is an independent observation interval, too many intervals may produce too many anamolies for the algorithm to reduce with sufficient accuracy. An upper limit on the number of intervals per frame time should be established.

BIBLIOGRAPHY

1. Black, Harold S. Modulation Theory. New York: D. Van Nostrand Company, Inc., 1953.

2. Forney, G. David, Jr. "The Viterbi Algorithm," Proceedings of the IEEE, Vol. 61, No. 3:268-278 (March 1973).

3. Hoel, Paul G., Sidney C. Port and Charles J. Stone. Introduction to Stochastic Processes. Boston: Houghton Mifflin Company, 1972.

4. Howard, Ronald A. Dynamic Probabilistic Systems, Volume I: Markov Models. New York: John Wiley and Sons, Inc., 1971.

5. Kleinrock, Leonard. Queuing Systems, Volume I: Theory. New York: John Wiley and Sons, Inc., 1975.

6. Matthews, Jerold C. and Carl E. Langenhop. Discrete and Continuous Methods in Applied Mathematics. New York: John Wiley and Sons, Inc., 1966.

7. Melsa, James L. and David L. Cohn. Detection and Estimation Theory. New York: McGraw-Hill Book Company, 1978.

8. Meer, David E. Phase Sequence Estimation for Laser Line-Scan Imagery in the Presence of Rayliegh Fading. Masters Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Dayton, Ohio, 45433, AFIT/GE/EE/79D-24.

9. Omura, Jim K. "On the Viterbi Decoding Algorithm," IEEE Transactions on Information Theory, Vol. IT - pp. 177-179 (January 1969).

10. Omura, Jim K. "Performance Bounds for Viterbi Algorithms," Proceeding of ICC - 81: IEEE, 2.2.1-2.2.5 (1981).

11. Papoulis, Athanasios. Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill, Inc., 1965.

12. Peebles, Peyton Z., Jr. Communication System Principles. Massachusettes: Addison-Wesley Publishing Company, 1976.

13. Sakrison, D. J. Communication Theory: Transmission of Waveforms and Digital Information. New York: John Wiley and Sons, Inc., 1968.

14. Scharf, Louis L., et. al. Modulo - 2π Phase Sequence Estimation, ONR Technical Report #27. Colorado State University, Fort Collins, Colorado, February 1978.

15. Schwartz, Mischa, et. al. Communication Systems and Techniques. New York: McGraw-Hill Book Company, 1966.

16. Shinghal, Rajjan and Godfried T. Toussaint. "The Sensitivity of the Modified Viterbi Algorithm to the Source Statistics," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 2, pp. 181-185 (March 1980).

17. Snyder, Donald L. Random Point Processes. New York: John Wiley and Sons, Inc., 1975.

18. Viterbi, A. J. "Error Bounds for Convolution Codes and an Asumptotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, Vol. IT-13, pp. 260-269, 1967.

19. Wozencraft, John M., and Irwin M. Jacobs. Principles of Communication Engineering. New York: John Wiley and Sons, Inc., 1975.

20. Ziemer, R. E. and W. H. Tranter. Principles of Communication Systems, Modulation, and Noise. Boston: Houghton Mifflin Company, 1976.

APPENDIX A

POISSON CHANNEL

POISSON CHANNEL:

The attempt and subsequent abandonment in modelling the channel as Poisson is presented in this appendix. The development centers around arriving at the branch metric of equation (53), using the parameters of a Poisson density. The discussion begins with the assumption that a low count rate, Poisson distribution, may be used to model the channel and concludes with the reasoning behind abandoning the Poisson model for low count rates.

The Poisson process is a counting process with a density function:

$$P\left\{N(T)=n\right\} = \frac{(\lambda T)^n e^{-\lambda T}}{n!} \qquad (68)$$

In words, equation (68) defines the probability that n counts or occurrences of some event will occur in time T. The parameter, λ is the rate of the process and has diminsions of events per unit time. When multiplied by the parameter T, the term (λ T) defines the mean or $E\left\{N\right\}$ of the process. Indeed, the rate parameter may be random, however, for this study, λ is assumed to be deterministic. Therefore, equation (68) may be rewritten as:

$$P\left\{N=n\middle|\lambda\right\} = \frac{(\lambda T)^n e^{-(\lambda T)}}{n!} \qquad (69)$$

What follows is a discussion of the reasoning behind the selection of the values used for the λT parameter.

A Poisson density for low count rates (λT) is not a continuous function. An example of such a density function is shown in Figure 60. The function exists only for integer values of n. As the rate or mean of the process increases, the delta functions of Figure 60 become more dense, thus approximating a continuous function. Therefore, to assume that the channel may be modelled as Poisson means that the occurrence of one "noise event" must not occur "too close," in time, to the next "noise event." Figure 61 may help to expand the terms, "noise event" and "too close." In expanding these two terms, an anology between the familiar RC low pass filter network and the denseness of the events will be drawn to add clarity.

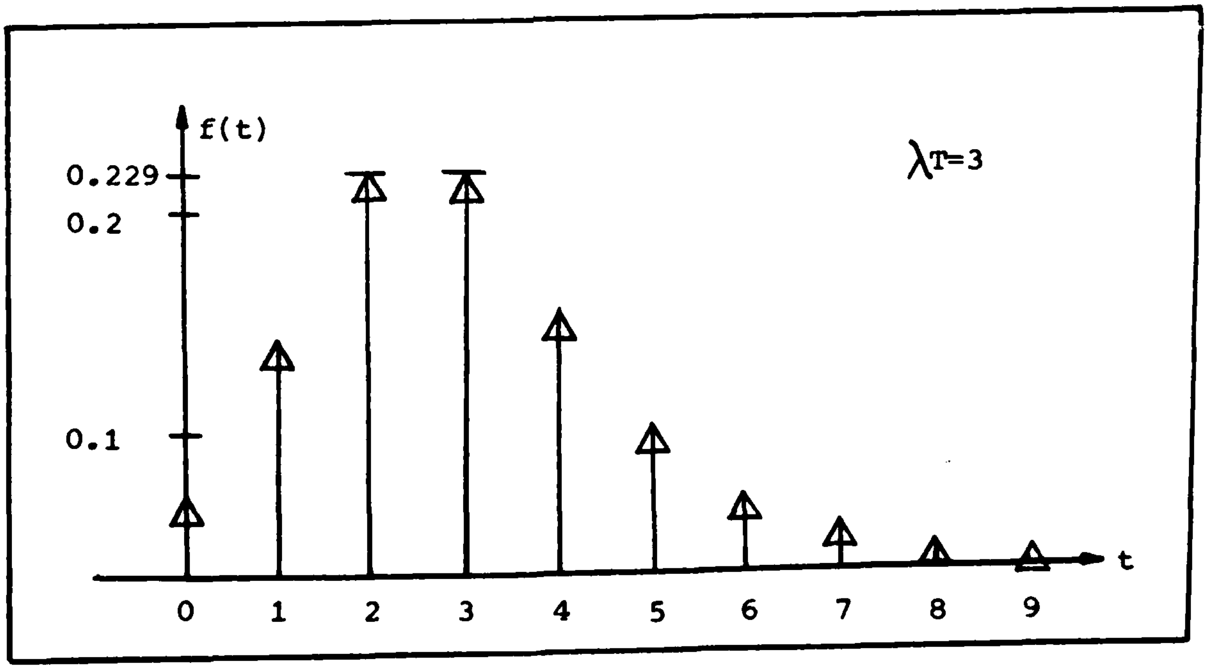


FIGURE 60. Sample Poisson Density Function - Low Count Rate

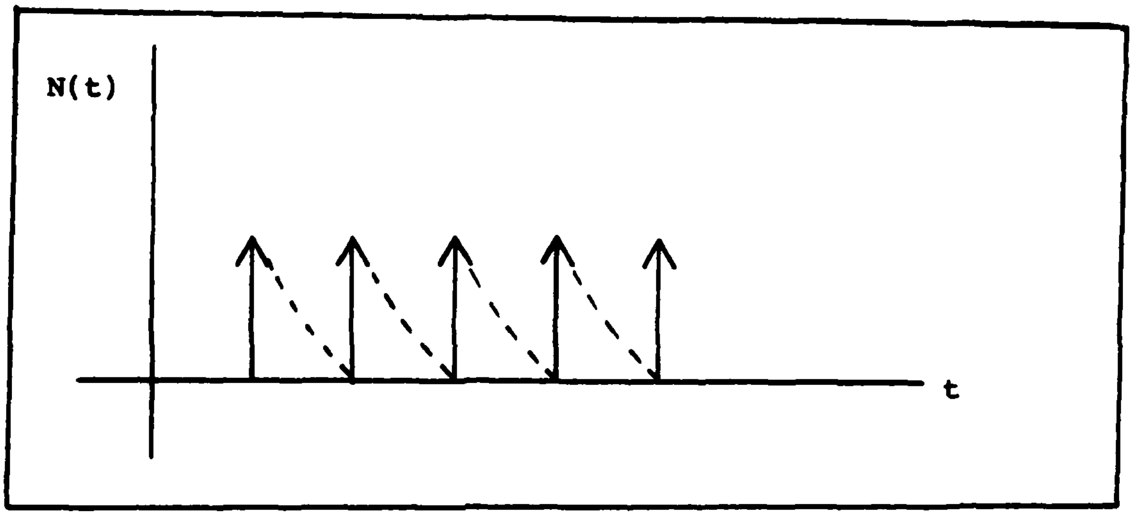


FIGURE 61A. Low Count Model

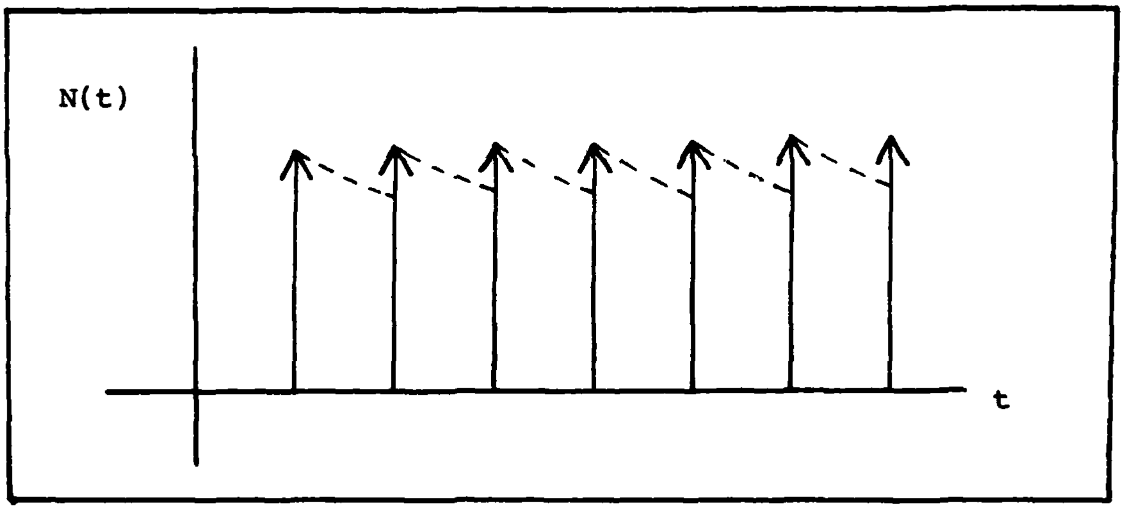


FIGURE 61B. High Count Model

Figure 61A depicts a noise process, N(t), and the delta functions represent a noise event. Since a Poisson process is a counting process, a noise event is not synonymous with amplitude such as is the case when using a Gaussian channel model. Instead, a collection of events (a count) over some interval (the bit interval previously defined) will constitute an "amplitude" value for the noise pulse. The dotted (exponential) line between each event represents the time necessary to keep the events far enough part in time, in order to keep the density function discrete. This line, and associated time is synonymous with the RC time constant of an RC, low-pass filter. The shape of the impulse response of a RC low-pass filter is dependent upon the values of R and C (resistance and capacitance). These values, in turn, determine the bandwidth of the filter.

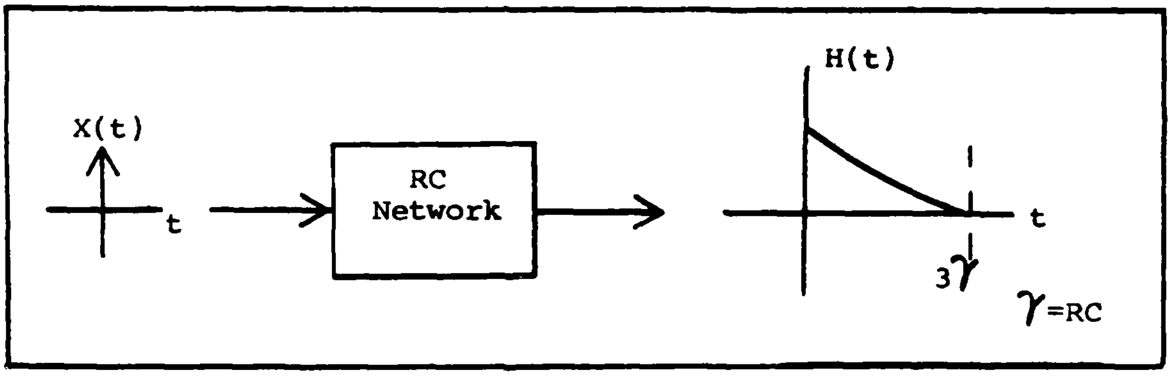


FIGURE 62. RC Impulse Response

Figure 62 depicts the impulse response of a RC Network. The effect of the impulse is said to diminish after 3γ, as depicted, where γ=RC is one time constant.

This same criteria was used as the basis for separation between noise events. That is, in order to ensure the probability density function is discrete, a "three time constant" separation was established. The events depicted in Figure 61B do not meet the criteria and, therefore, would describe a process with events too dense to be described by a discrete distribution.

Since the values for R and C also determine the bandwidth of the frequency response of the RC network, the bandwidth of the receiver may be used to equate the signal and receiver parameters to the count rate of the channel model.

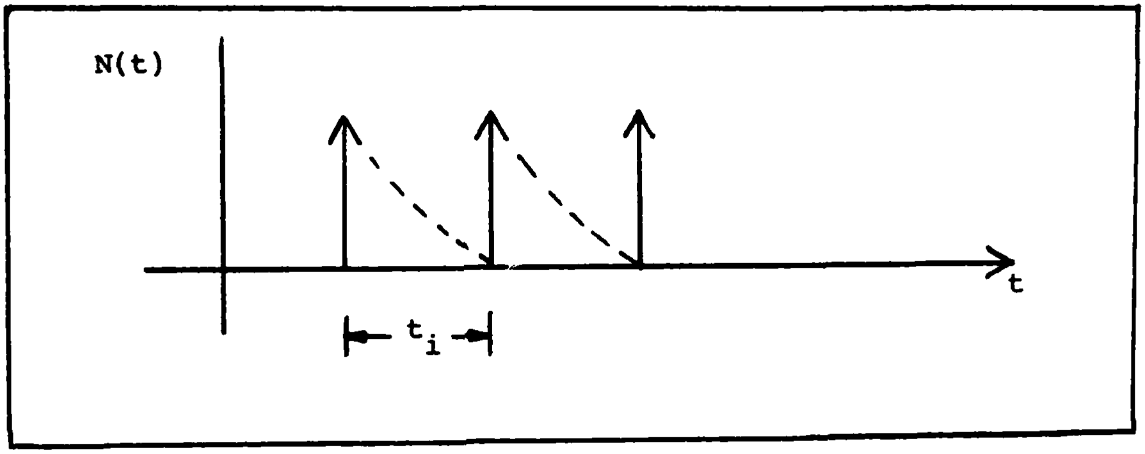


FIGURE 63. Inter-occurrence Time

Figure 61A has been redrawn in Figure 63 defining the inter-occurrence time, t_i, between noise events. By the discussion above, clearly:

$$t_i \geq 3\gamma = 3\ (RC) \qquad (70)$$

The bandwidth of the receiver will be defined as:

$$B \overset{\sim}{\geq} \frac{1}{2t_r} \quad \text{(Ref 20)} \qquad (71)$$

where $\overset{\sim}{\geq}$ stands for "greater than, but proportional to." The term, t_r is the rise time of the expected pulse. Using the definitions presented earlier:

T = frame (sample) time

t_b = bit interval time

M = # of bit intervals per frame

clearly:

$$t_b = T/M \tag{72}$$

and pulse rise time may be defined as:

$$t_r = K\, t_b \qquad K \ll 1 \tag{73}$$

The bandwidth, B_{RC}, of the RC network is given by:

$$B_{RC} = \frac{1}{\gamma} = \frac{1}{RC} \tag{74}$$

Disregarding the constants of proportionality in equations (71), and (71) equated with (74) to give:

$$\frac{1}{\gamma} = \frac{1}{2t_r} \approx B \tag{75}$$

Since from Figure 63:

$$t_i \geq 3\gamma \tag{76}$$

and by substitution:

$$\frac{1}{2Kt_b} = B_{RC} = \frac{1}{\gamma} \tag{77}$$

Since, t_b is known, a value for γ is available and, therefore, a minimum value for t_i is also obtainable.

However, the average inter-occurrence time, $\langle t_i \rangle$, defines the rate parameter λ by:

$$\lambda = \frac{1}{\langle t_i \rangle} \tag{78}$$

Assuming only a small difference in values for t_i for all i:

$$\langle t_i \rangle \approx t_i \qquad (79)$$

The average inter-occurrence time will define the rate parameter. Now, based on the "bandwidth" of the receiver, a value for the Poisson channel rate parameter is available by combining equations (76), (77), and (78).

Since each bit interval is an independent observation interval, a range specifying the mean of the counting process was established on a bit interval basis. That is:

$$5 \leq \lambda t_b \leq 50 \qquad (80)$$

This range is arbitrary, but such that a Poisson process with these mean values is still considered discrete. From this range and equaticns (77) and (78), a range for receiver bandwidth was obtained:

$$24\text{KHz} \leq B \leq 240\text{KHz} \qquad (81)$$

This range was considered realistic, however, its accuracy was not determined nor considered.

It is now convenient to introduce the problems in implementation using the Poisson channel model for low count rates.

First, the signal process is not a counting process. It, therefore, becomes difficult to equate a "counting threshold" to a receiver amplitude threshold based on the expected signal pulse amplitude. A counting threshold was needed to define the occurrence of an anamolous pulse within a bit interval. It was decided that a count within a bit interval greater than some value P, would constitute the occurrence of an anomoly. But this threshold has no meaning with respect to an amplitude threshold that is assumed to guarantee a probability of signal pulse detection equal to one.

If such a correlation could be determined, the discrete channel model of Figure 9 could be used to derive the path metric term:

$$-\ln\left\{P(\underline{Z}|\underline{X}_j)\right\} \qquad j=1,2,\ldots.M \qquad (82)$$

Since each bit interval is independent, equation (82) becomes:

$$-\ln\left\{\prod_{j=1}^{M}\left\{P\left\{Z_j|X_i\right\}\right\}\right\} \qquad (83)$$

Instead of the distance measure for the Gaussian case (equation (51)), this term is specified by a product of the bit crossover probabilities. The distance measure for each branch, in the Gaussian case, is obtained by comparing the receiver vector $\underline{Z}$, with each possible input vector $\underline{S}^i$. For the Poisson case, this distance measure becomes a pairwise product of crossover probabilities between the observed $\underline{Z}$ vector and each possible signal vector. An example is most beneficial.

Assuming a four element vector (i.e., 4 bit intervals per frame), one pairwise product would be that of equation (85) for the given received vector and signal vector:

$$\underline{Z} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad \underline{S}^i = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{84}$$

$$\prod_{i=1}^{4} \left\{ P(Z_i | X_i) \right\} = P(0|0) \cdot P(1|1) \cdot P(1|0) \cdot P(0|0) \tag{85}$$

Unfortunately, the crossover probabilities are a function of the count threshold as each crossover probability describes the net affect of the channel and receiver.

Another problem with the Poisson channel had to do with sensitivity to the counting threshold, P. For low count rates, the Poisson distribution exists only for integer values of event occurrences, n. Therefore, the count threshold must be some integer value. Additionally, the integer value must be based on the mean value (λT) in equation (69). The counting threshold will produce the $\langle$ZCR$\rangle$ needed to specify the performance of the receiver (algorithm).

Numerous thresholds for various distribution functions defined by the range of equation (80) were tested. Values for P were chosen for counts less than, equal to, and greater than the mean of the distribution. Particularly for distributions defined by the low end of equation (80), there was little control over the number of anamolies produced per frame. A change in the value of P by ±1 produced a large change in the average number of anamolies per frame. Not only did this phenomenon make a study of algorithm performance based on changes in $\langle$ZCR$\rangle$ impossible, it also implied that a low count rate Poisson channel may not be appropriate. Hence, for the two reasons cited, a nonzero mean Gaussian density was chosen to model the channel. For random variates within $\pm 3\sigma$ of the mean, the Gaussian density function describes accurately a high count rate Poisson distribution. The simulations are represented, however, under the assumptions of a Gaussian channel.

APPENDIX B

SAMPLES OF QUANTIZED VERSIONS OF THE ASSUMED MESSAGE SET

SAMPLES OF MESSAGE SEQUENCES:

Samples of quantized versions of the assumed message set are presented in this appendix. While no estimations of signal numbers 1, 2, 4, or 5 were performed, these signals were used to generate signal numbers 7, 8, and 9. The equations used to provide each sample are provided for reference. For each expression, the rate parameter, λ, is defined as:

$$\lambda = \exp \left\{-u\right\}$$

where u is a uniform, (0, 1), random variable.

SIGNAL #	GENERATING EXPRESSION
$M_1(t)$	$1.0-\mathrm{EXP}\left\{-\lambda t\right\} \quad 0 \leq t \leq 10.0$
$M_2(t)$	$\mathrm{EXP}\left\{-\lambda t\right\} \quad 0 \leq t \leq 10.0$
$M_3(t)$	$.5+\left\{.5*\mathrm{SIN}(\lambda t-\lambda)\right\} \quad 0 \leq t \leq 10.0$
$M_4(t)$	$1.0-\mathrm{EXP}\left\{\lambda(t-10)\right\} \quad 0 \leq t \leq 10.0$
$M_5(t)$	$\mathrm{EXP}\left\{\lambda(t-10)\right\} \quad 0 \leq t \leq 10.0$
$M_6(t)$	$\mathrm{EXP}\left\{-(\lambda(t-5))^2/10\right\} \quad 0 \leq t \leq 10.0$
$M_7(t)$	$M_1(2t) \quad 0 \leq t \leq 5.0$
	$M_4(2(t-5)) \quad 5.0 \leq t \leq 10.0$
$M_8(t)$	$M_1(2t) \quad 0 \leq t \leq 5.0$
	$M_2(2(t-5)) \quad 5.0 \leq t \leq 10.0$
$M_9(t)$	$M_5(2t) \quad 0 \leq t \leq 5.0$
	$M_4(2(t-5)) \quad 5.0 \leq t \leq 10.0$
$M_{10}(t)$	$\mathrm{EXP}\left\{-\lambda \lvert t-5 \rvert\right\} \quad 0 \leq t \leq 10.0$

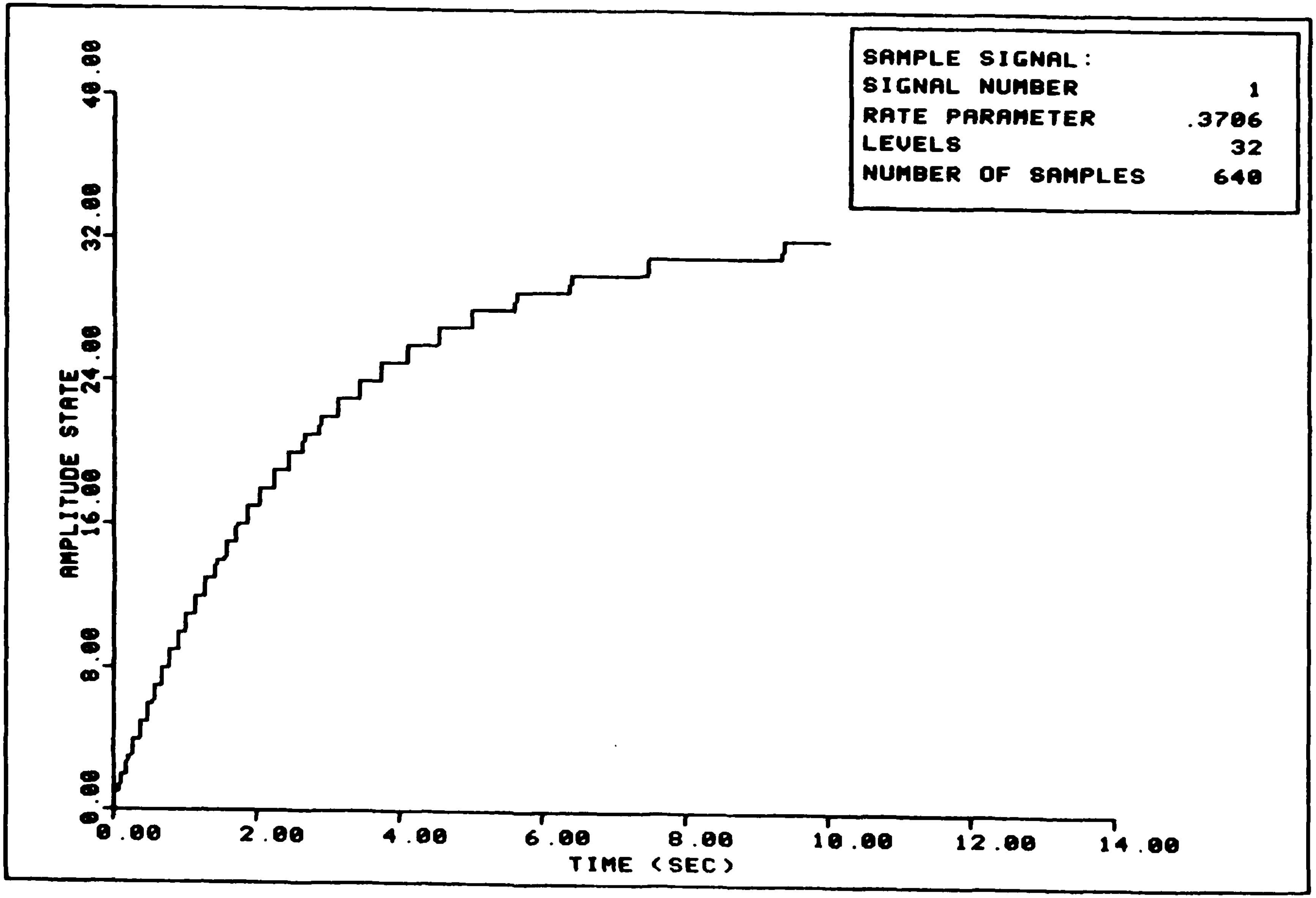


FIGURE 64. Sample Message Sequence

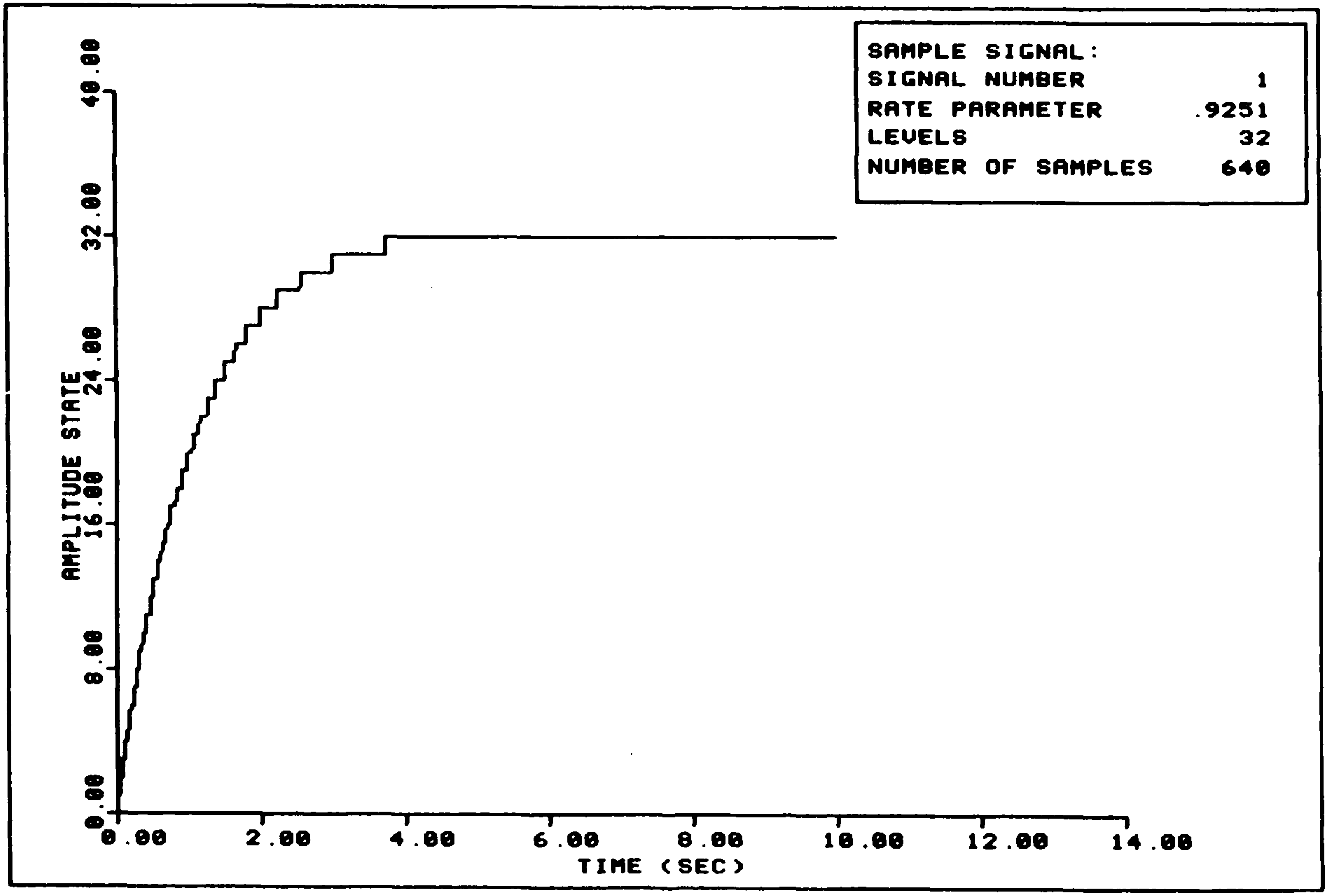


FIGURE 65. Sample Message Sequence

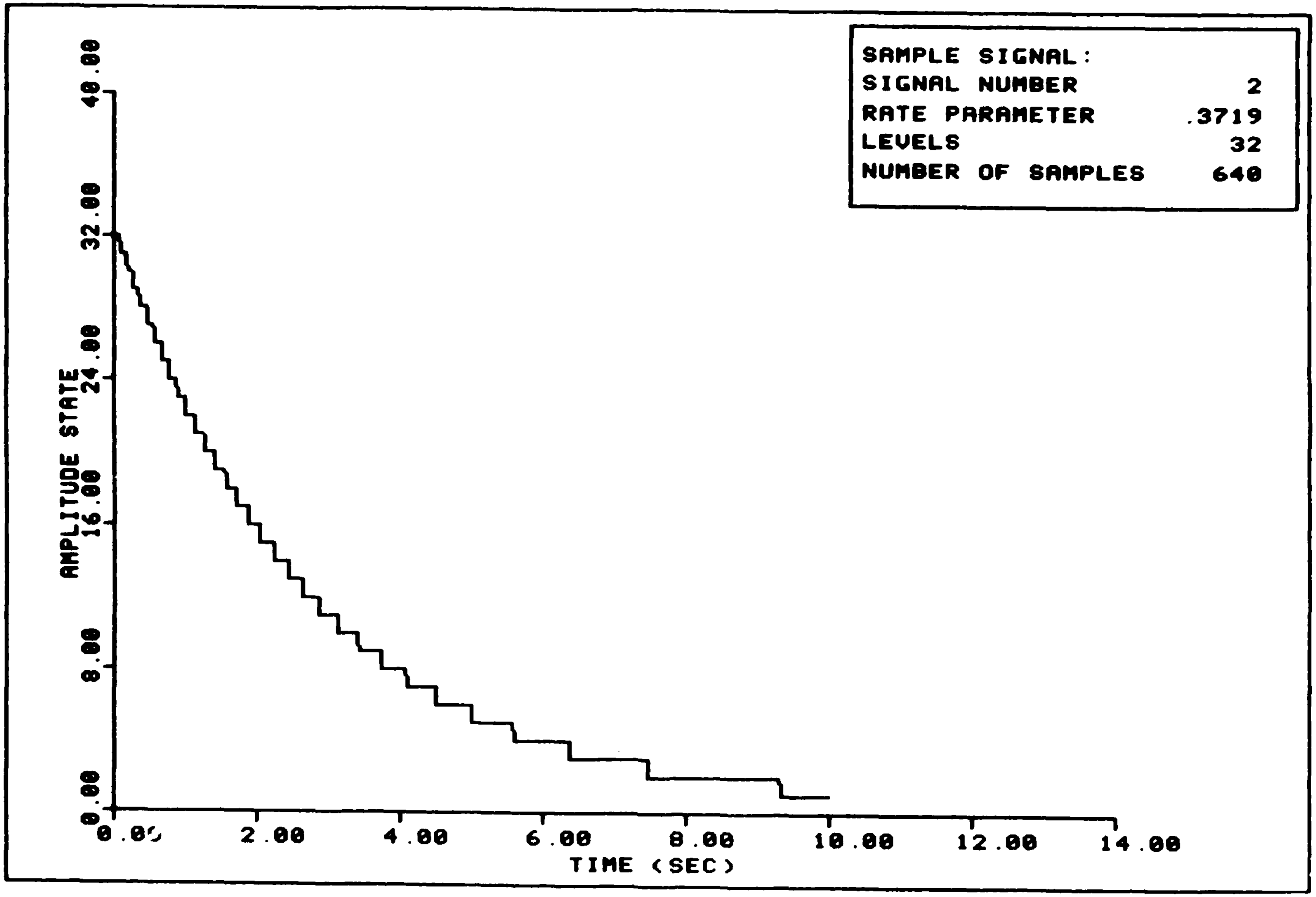


FIGURE 66. Sample Message Sequence

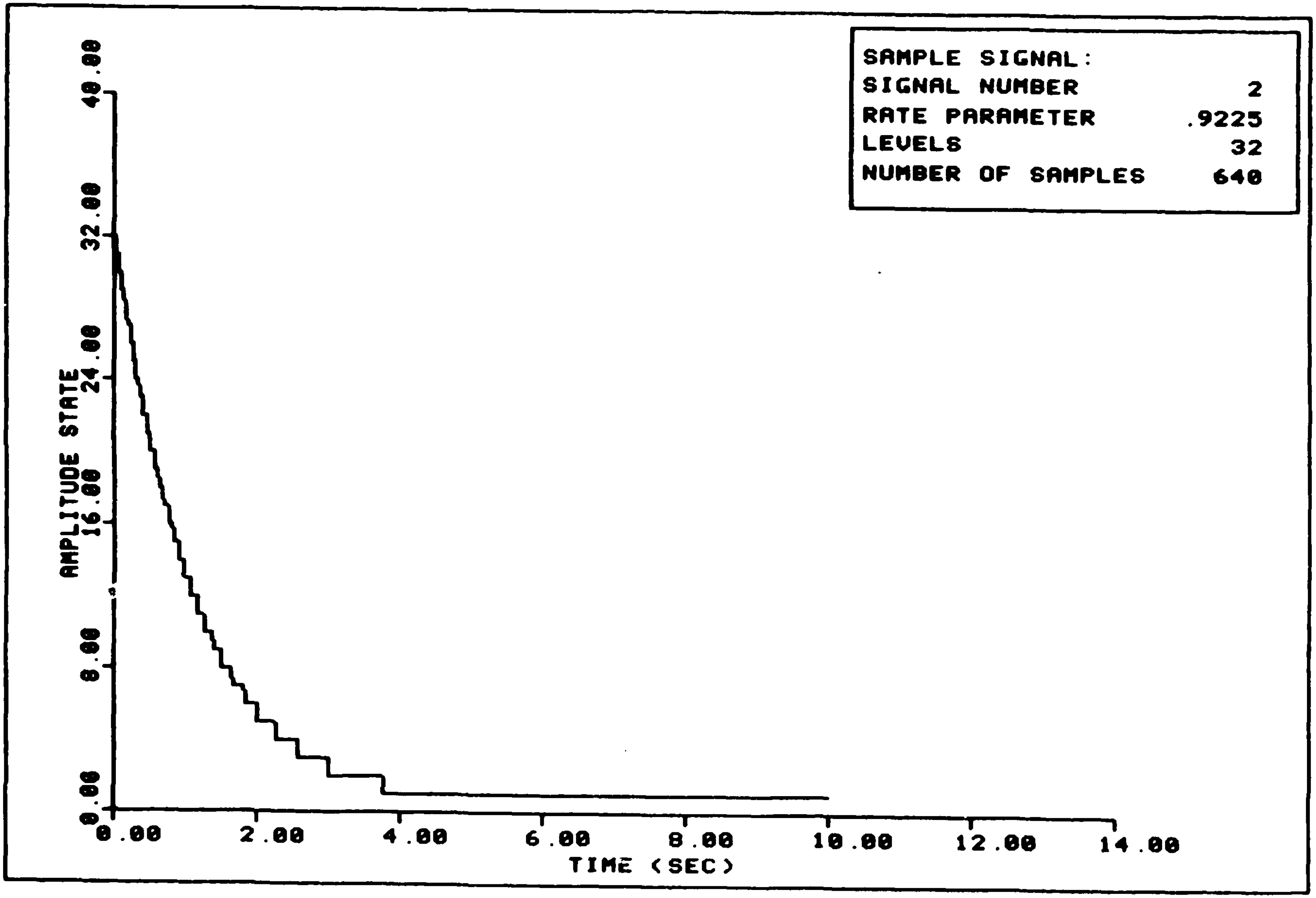


FIGURE 67. Sample Message Sequence

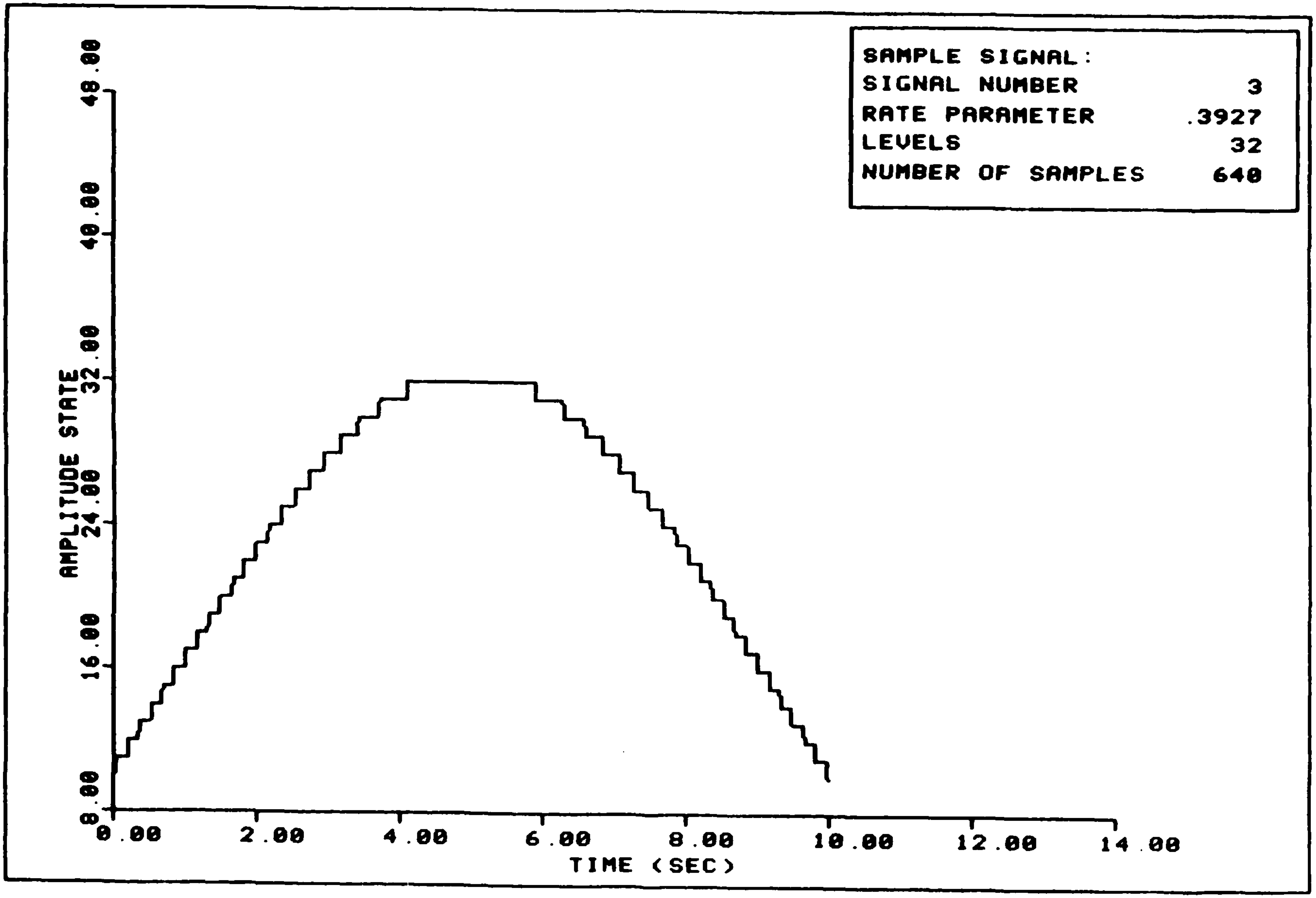


FIGURE 68. Sample Message Sequence

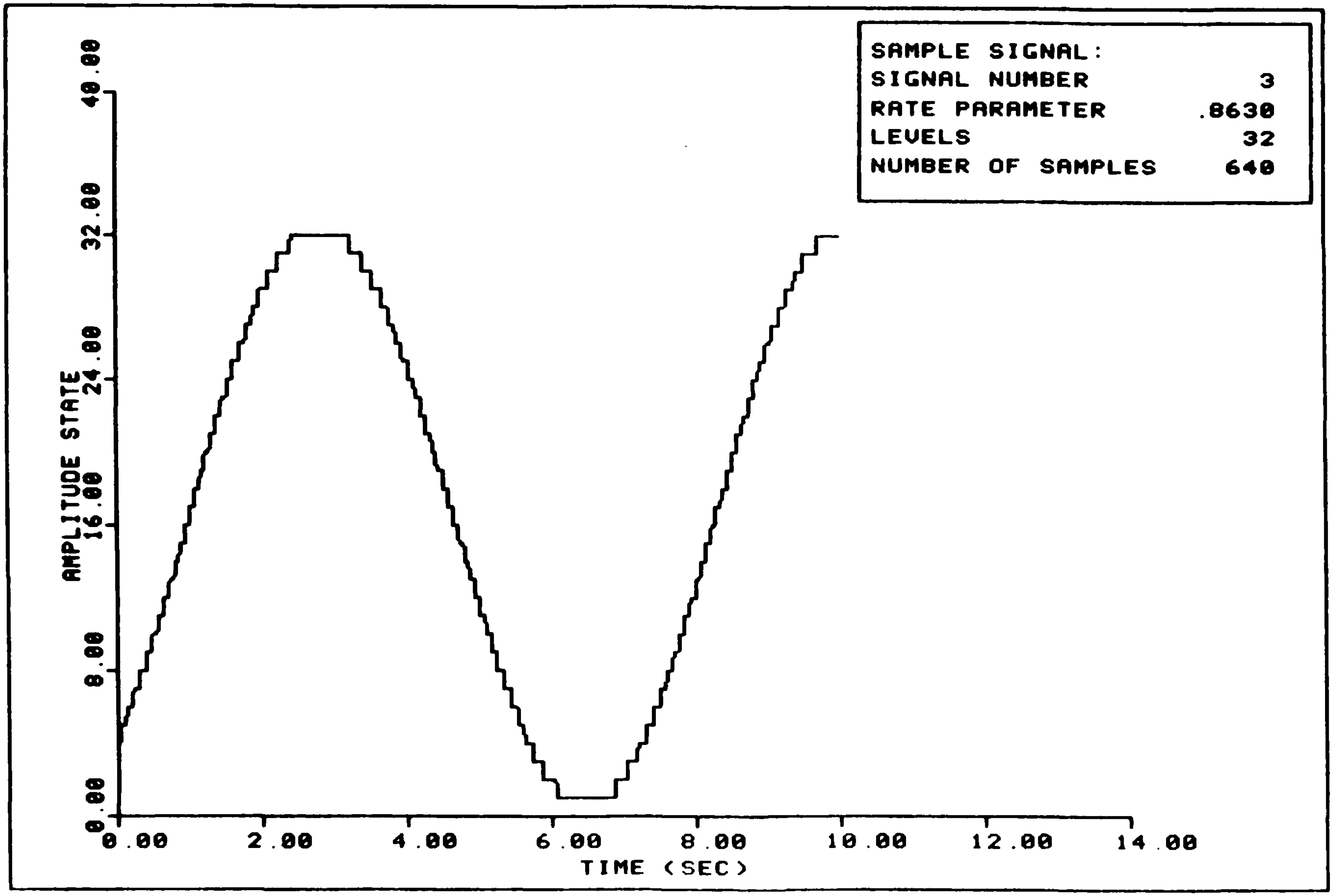


FIGURE 69. Sample Message Sequence

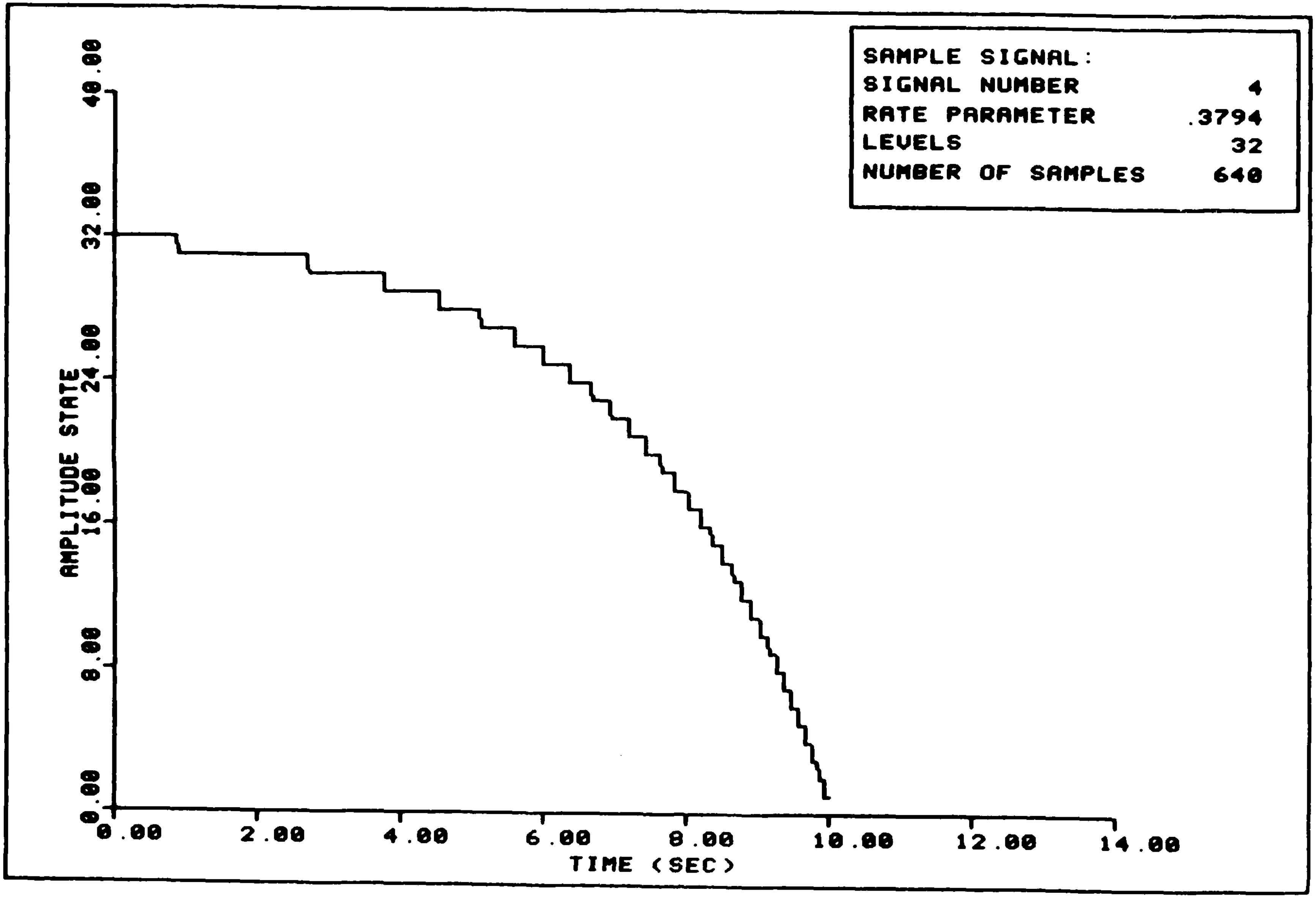


FIGURE 70. Sample Message Sequence

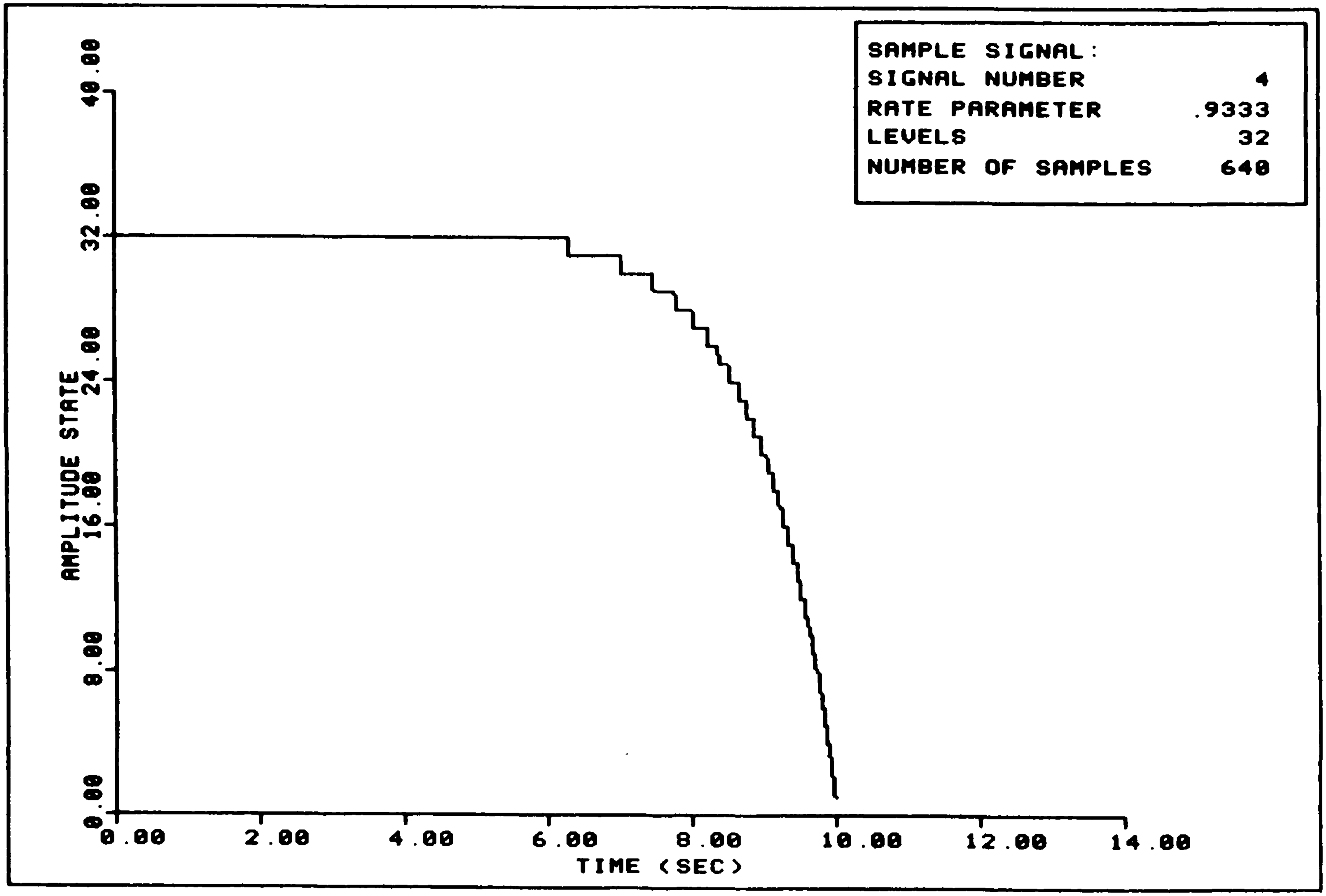


FIGURE 71. Sample Message Sequence

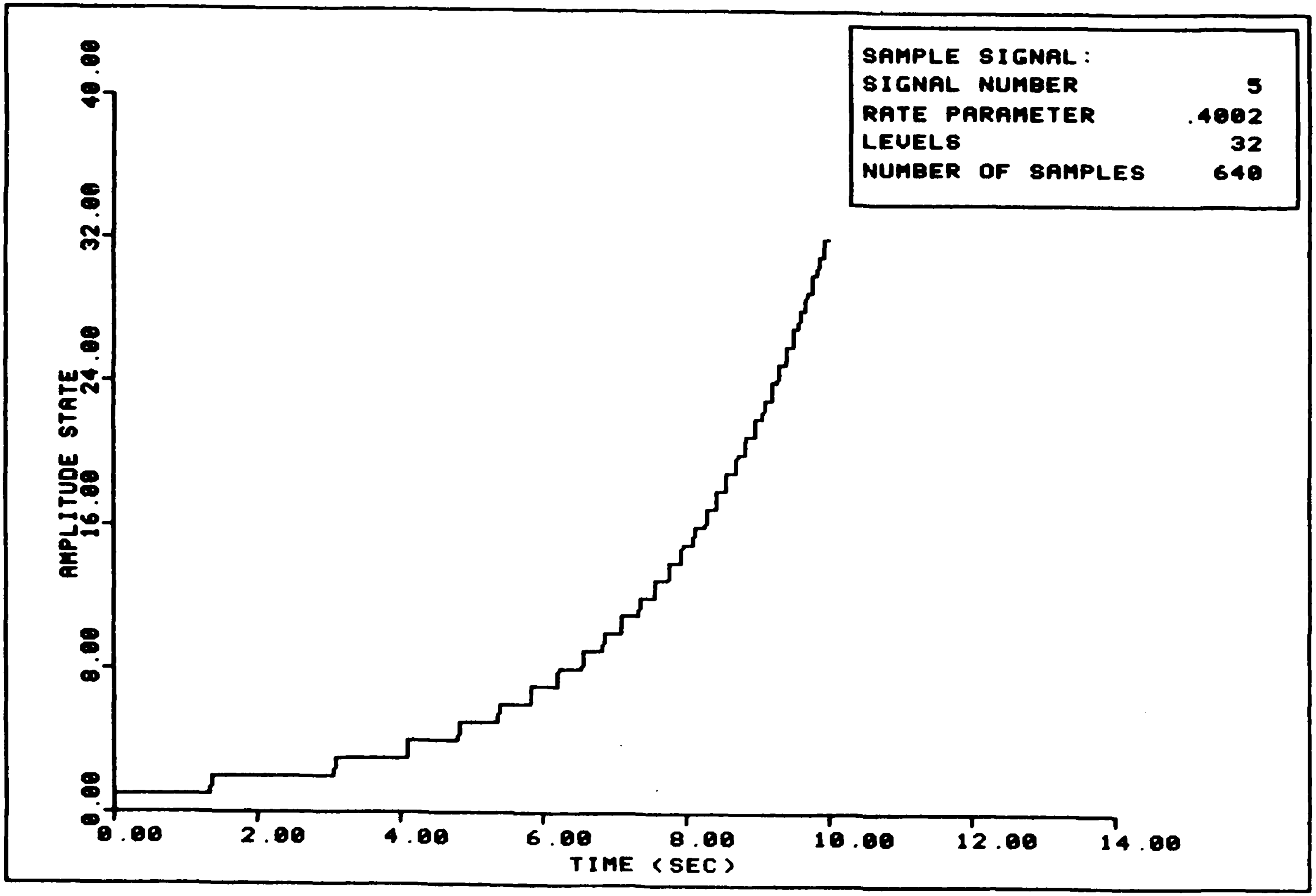


FIGURE 72. Sample Message Sequence

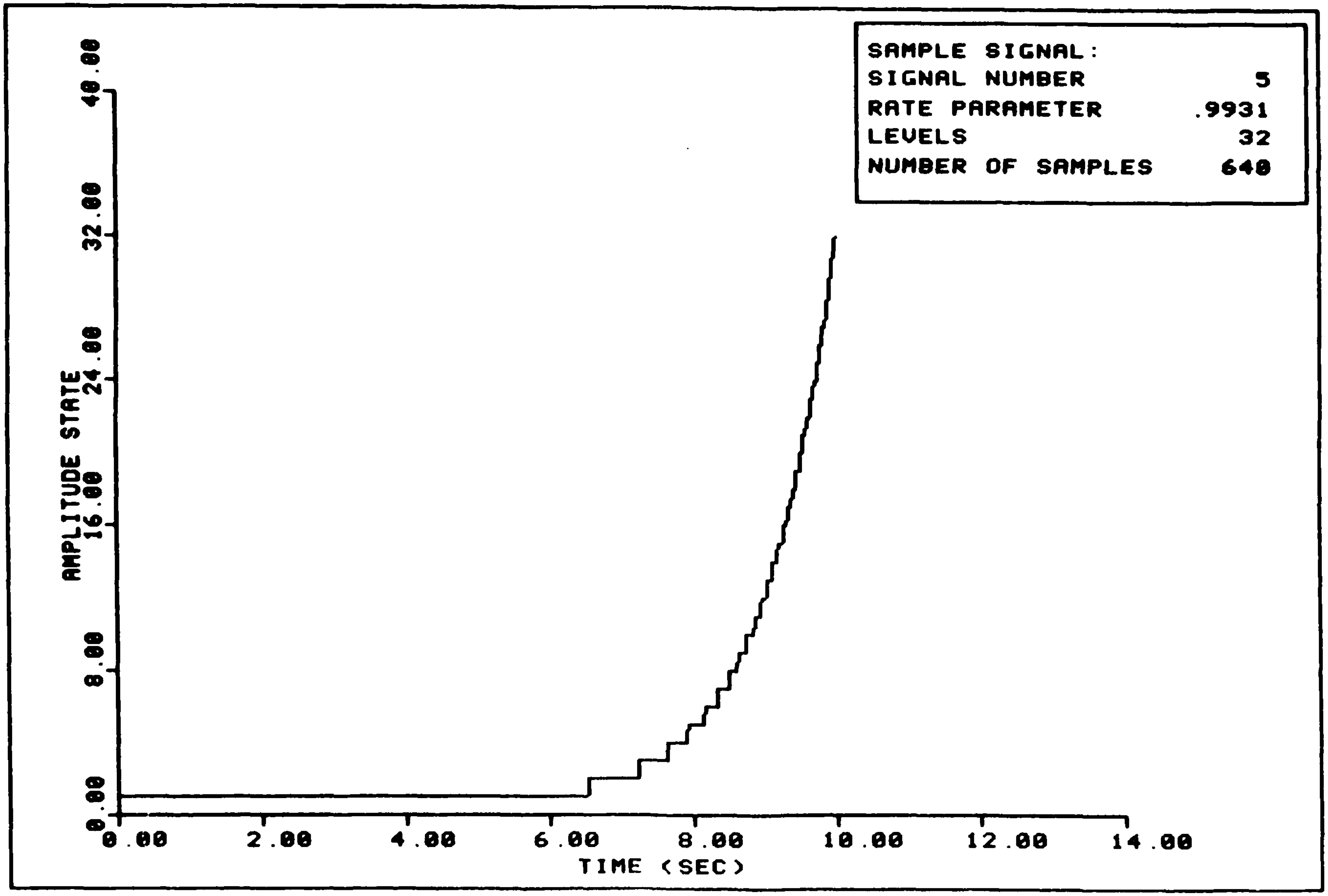


FIGURE 73. Sample Message Sequence

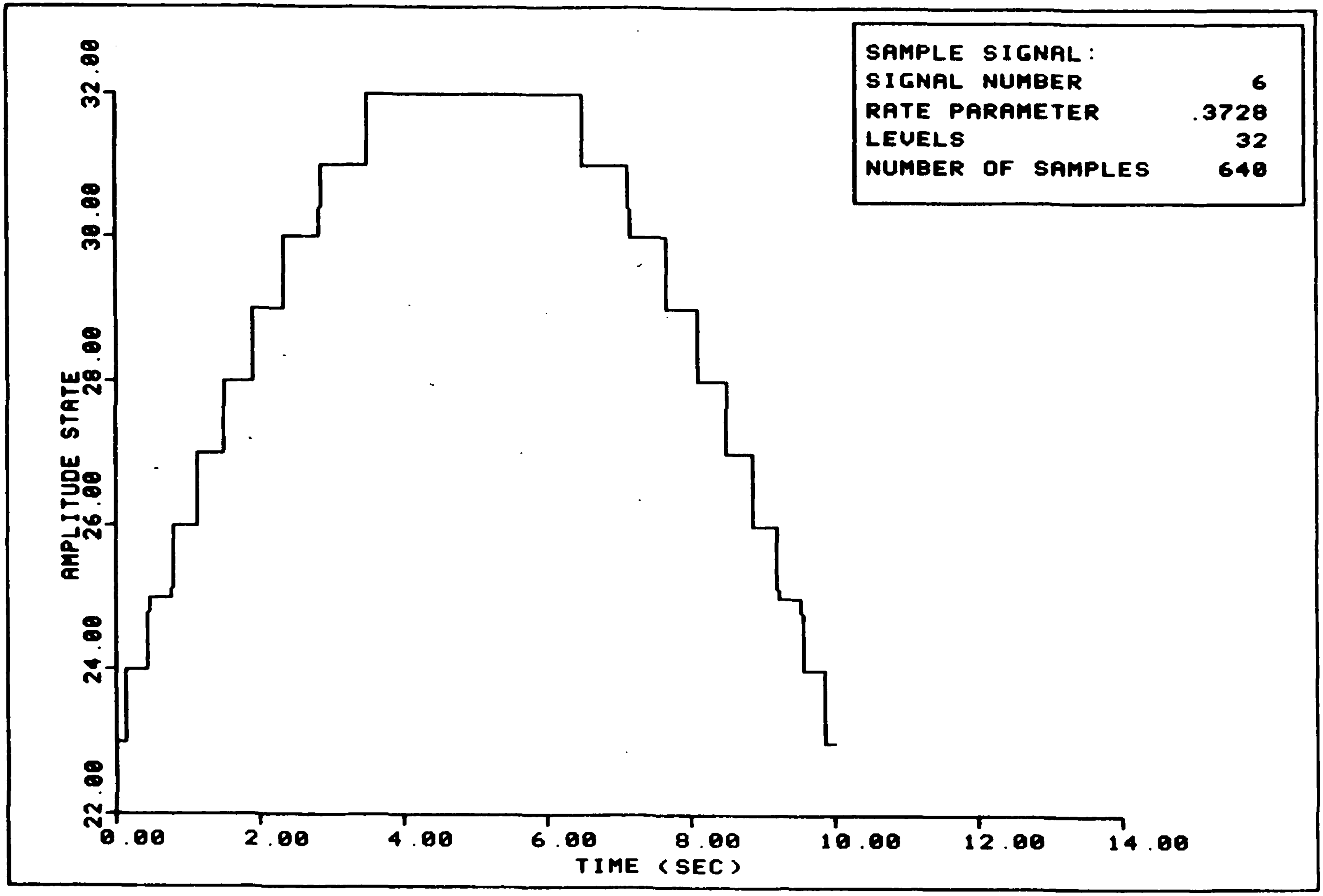


FIGURE 74. Sample Message Sequence

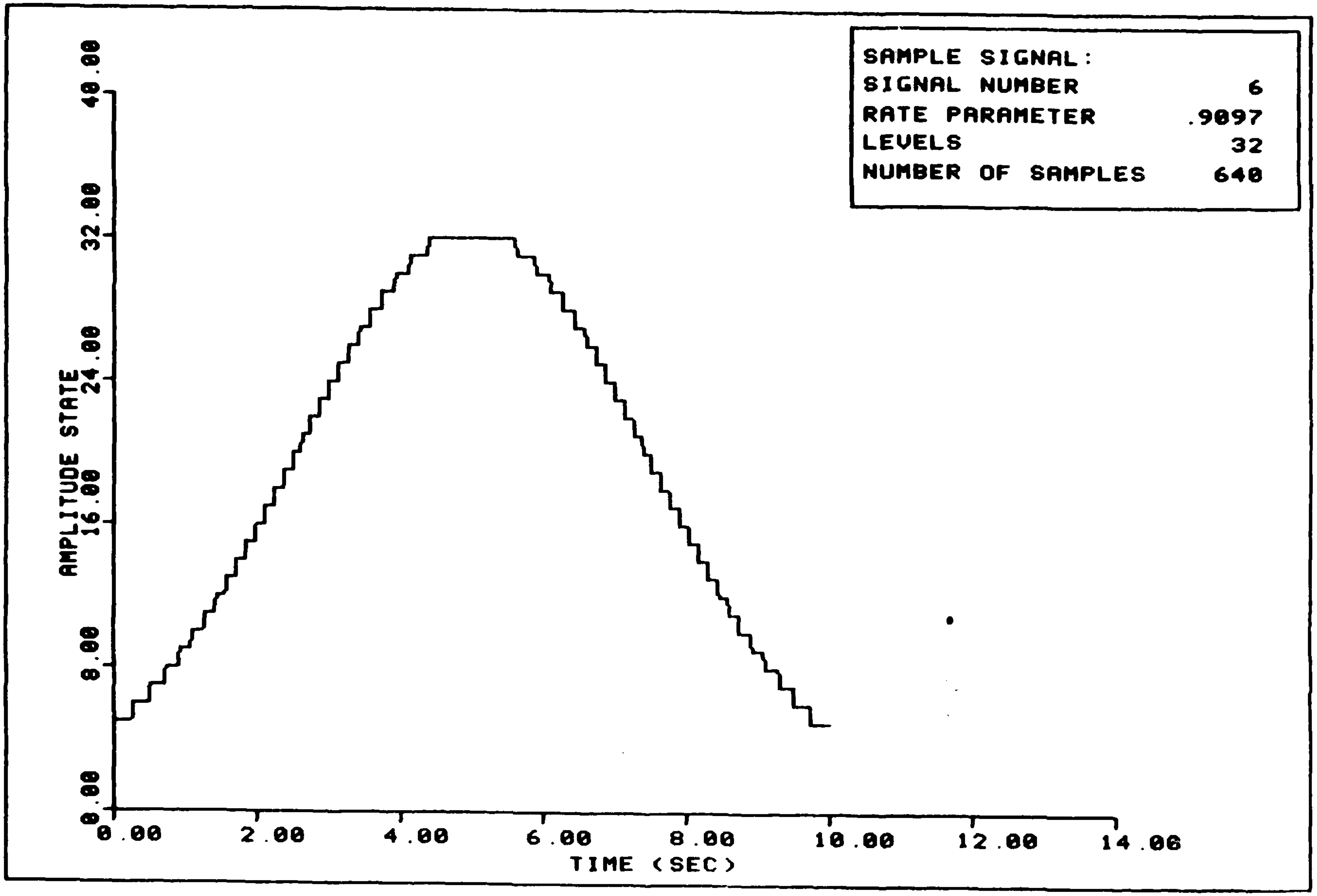


FIGURE 75. Sample Message Sequence

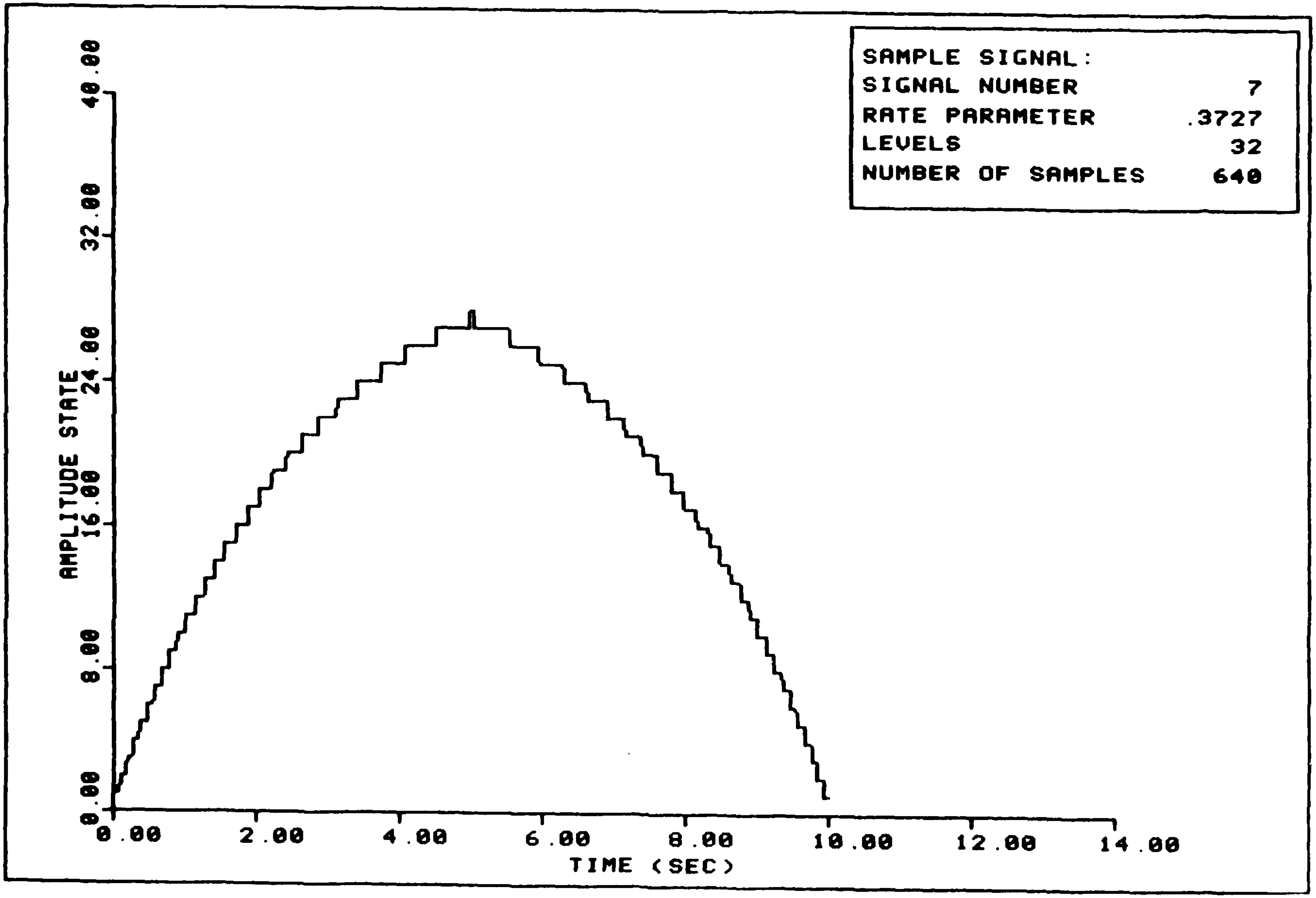


FIGURE 76. Sample Message Sequence

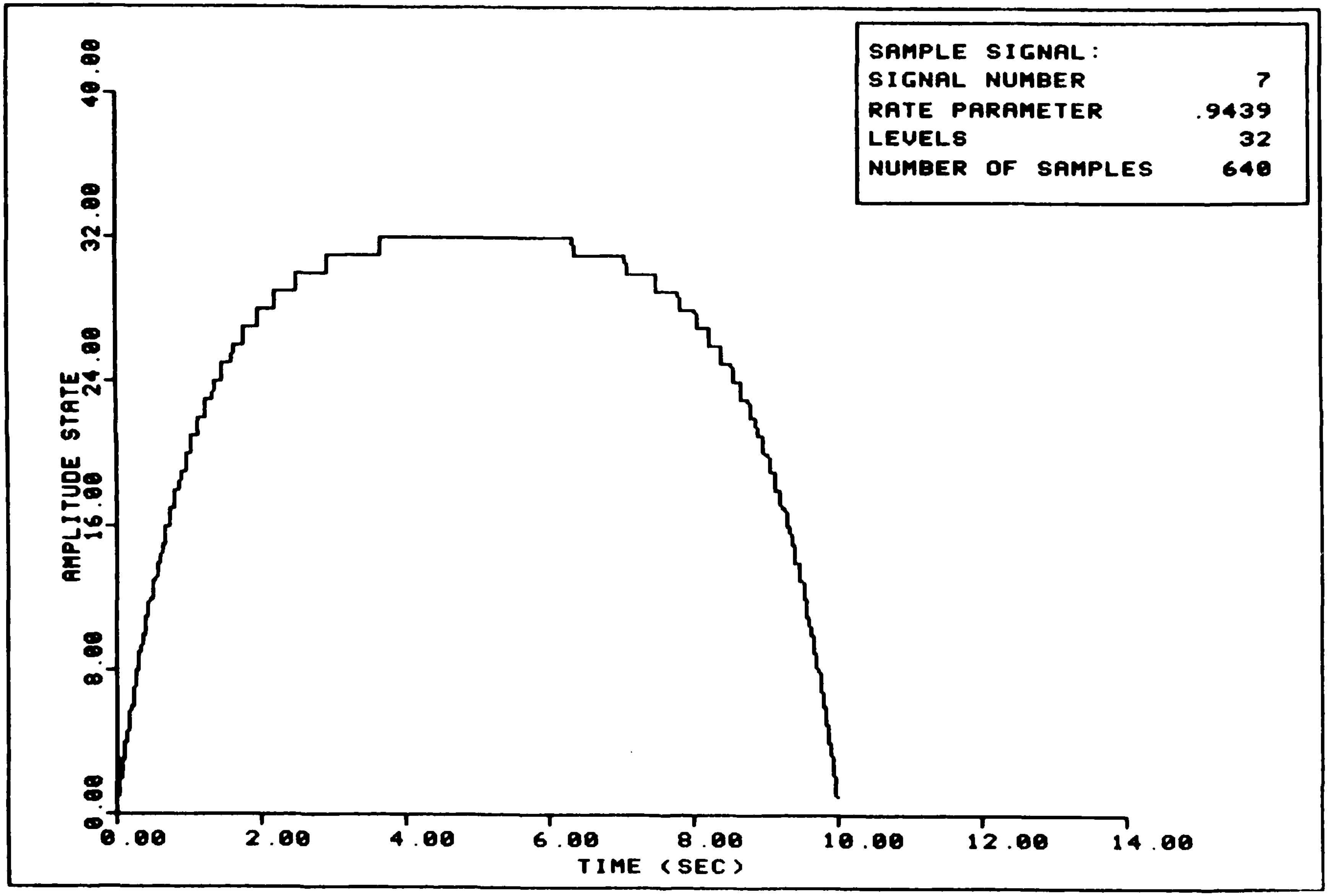


FIGURE 77. Sample Message Sequence

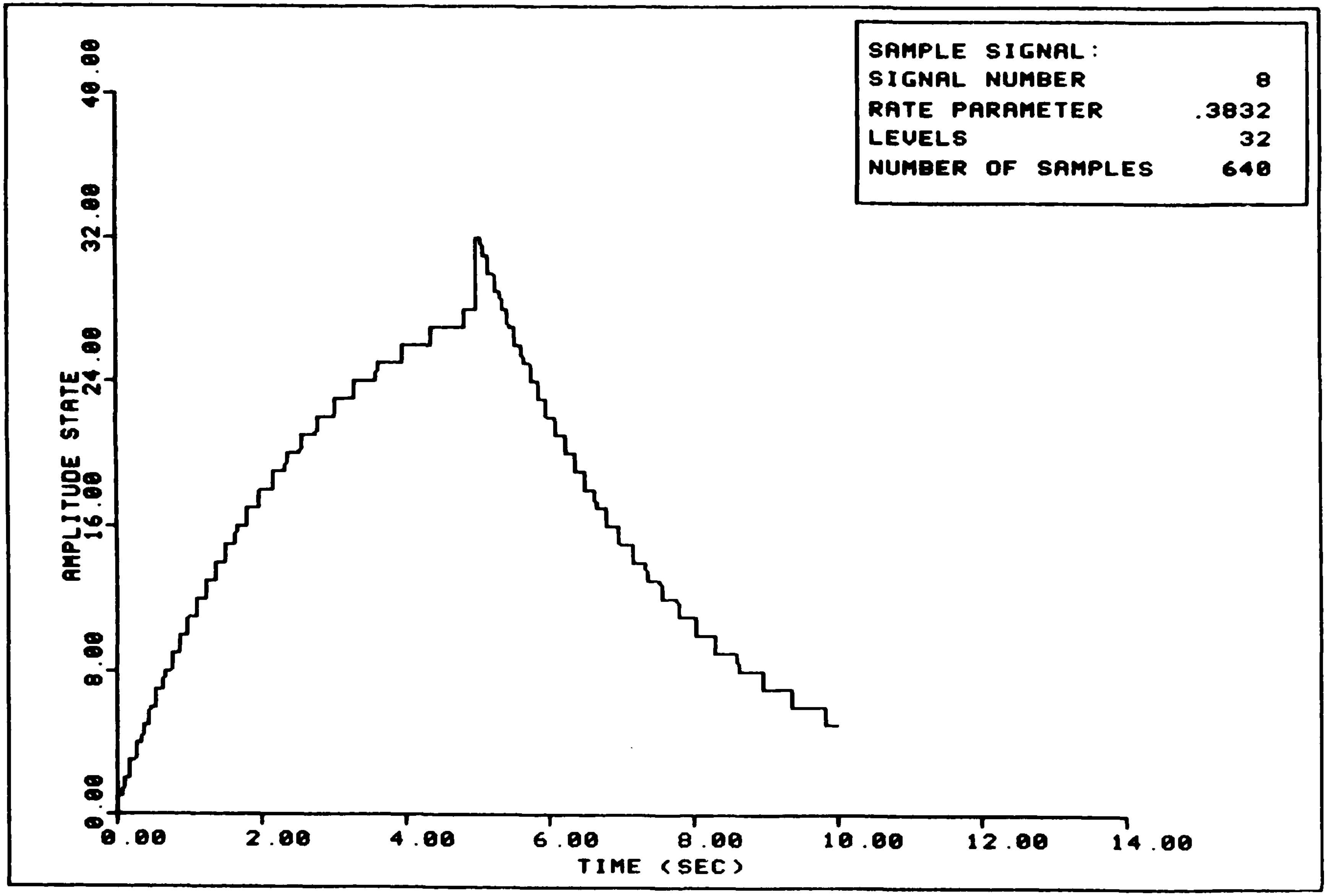


FIGURE 78. Sample Message Sequence

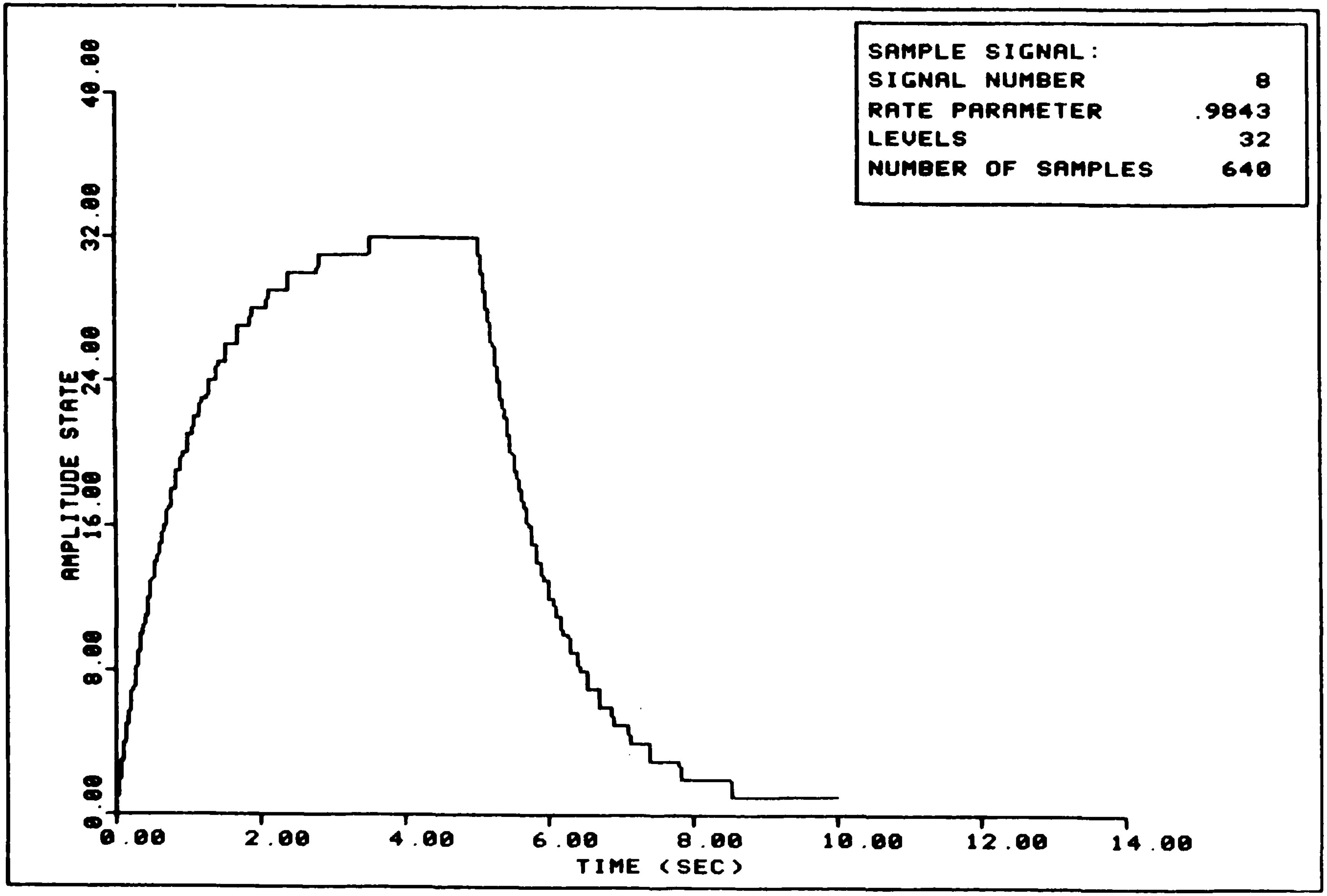


FIGURE 79. Sample Message Sequence

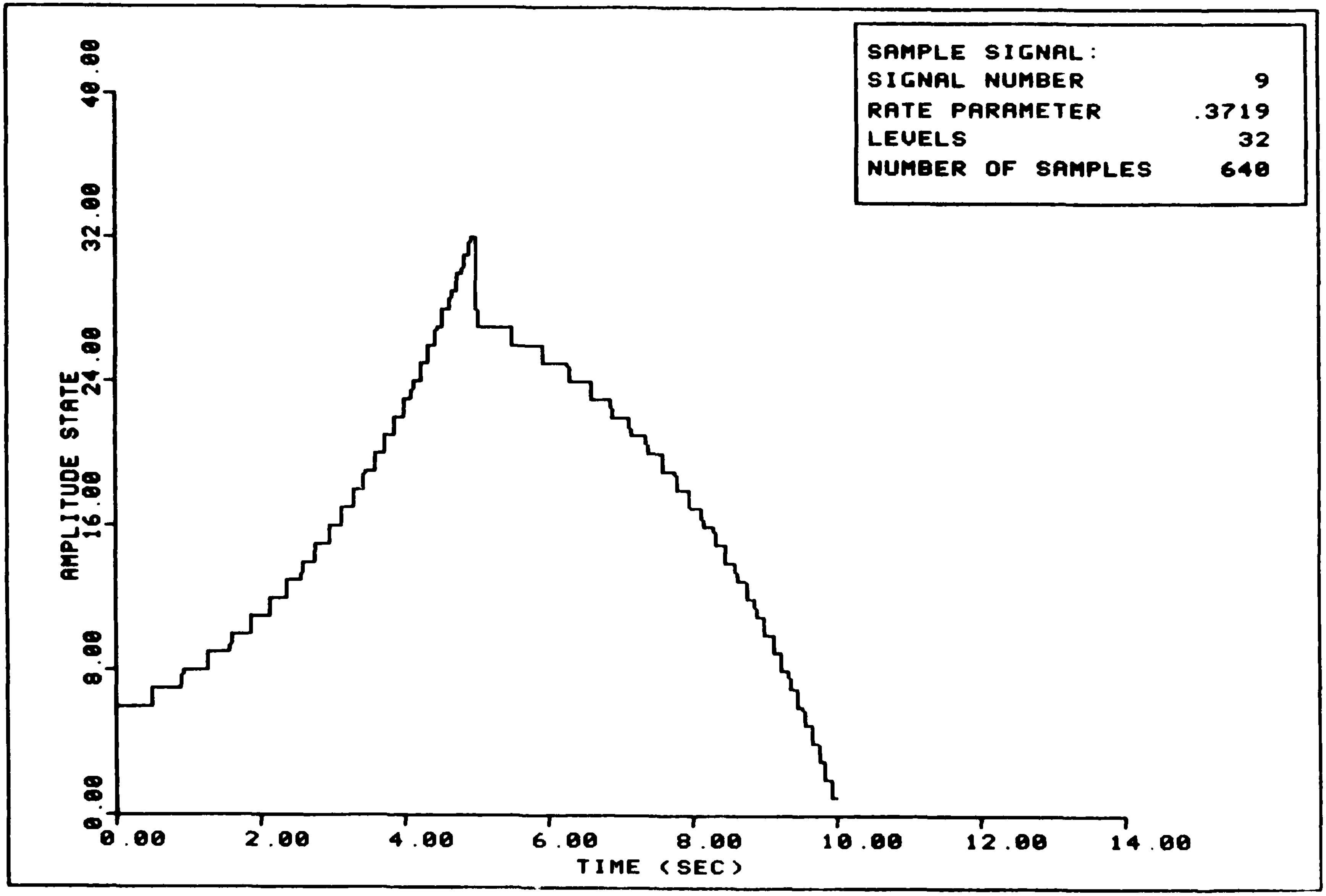


FIGURE 80. Sample Message Sequence

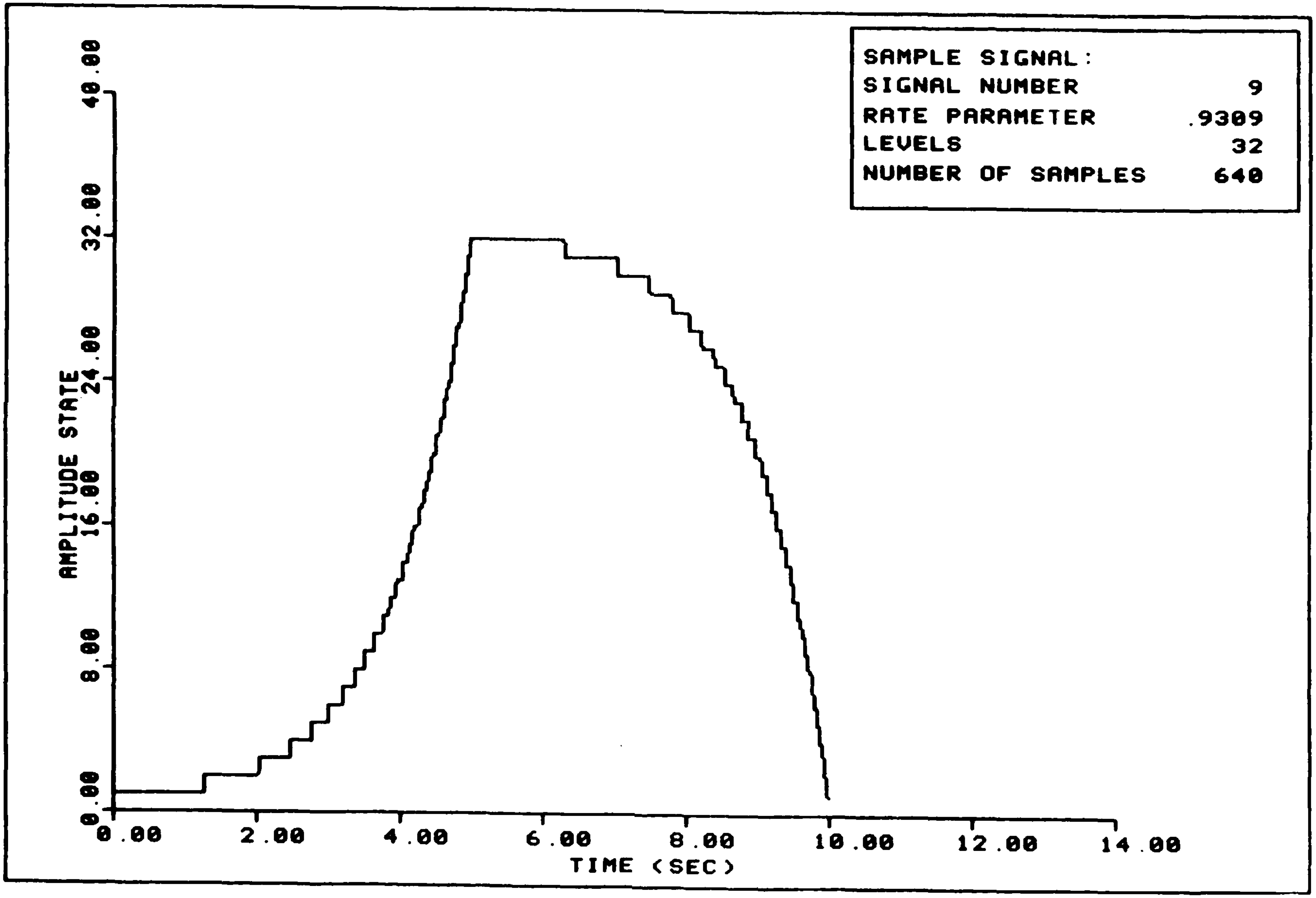


FIGURE 81. Sample Message Sequence

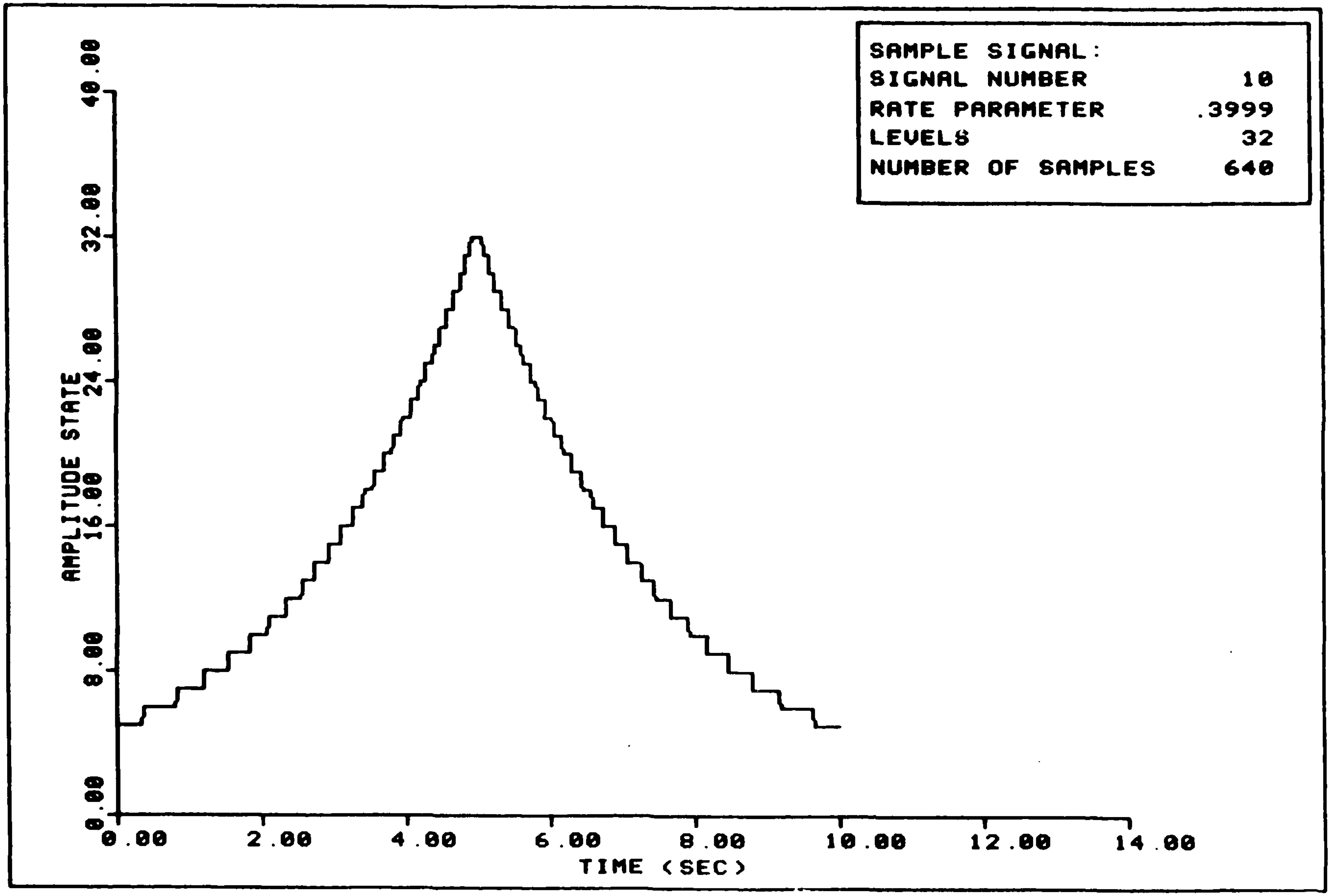


FIGURE 82. Sample Message Sequence

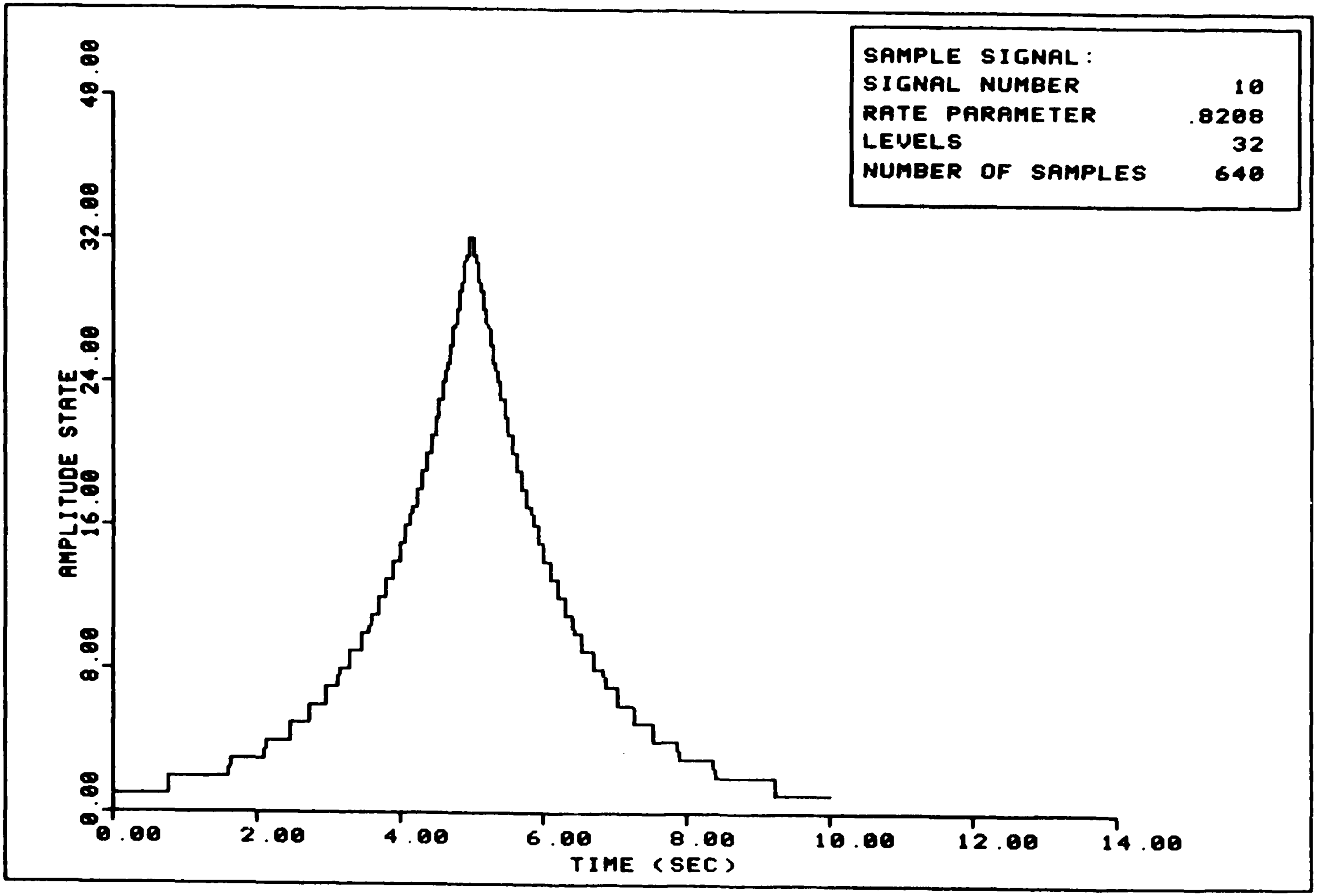


FIGURE 83. Sample Message Sequence

APPENDIX C

SAMPLE ESTIMATIONS

SAMPLE ESTIMATIONS:

Sample estimations for signal numbers 3, 7, 8, 9, and 10 are presented in this appendix. The appendix is divided into five sections, corresponding to each of the above mentioned signals. For each signal, sample estimations and ensemble performance graphs are provided for different $\langle ZCR \rangle$ values. The $\langle ZCR \rangle$ value corresponds to three categories of performance:

(1) Confident reconstruction of the entire ensemble.

(2) Confident reconstruction of only a portion of the ensemble (i.e., marginal reconstruction for entire ensemble with reference to test).

(3) Questionable reconstruction of most or all of the ensemble.

These three categories of performance occur for $\langle ZCR \rangle$ values of 18, 23, and 26 for all the signals except #3. For signal #3, the estimator performance is questionable for $\langle ZCR \rangle = 23$ and marginal for a $\langle ZCR \rangle$ value greater than 18 and less than 23. Since the estimator performance degrades rapidly for $\langle ZCR \rangle > 18$, sample estimations for $\langle ZCR \rangle = 18$ and $\langle ZCR \rangle = 23$ are provided, and only ensemble performance curves presented for $\langle ZCR \rangle = 26$.

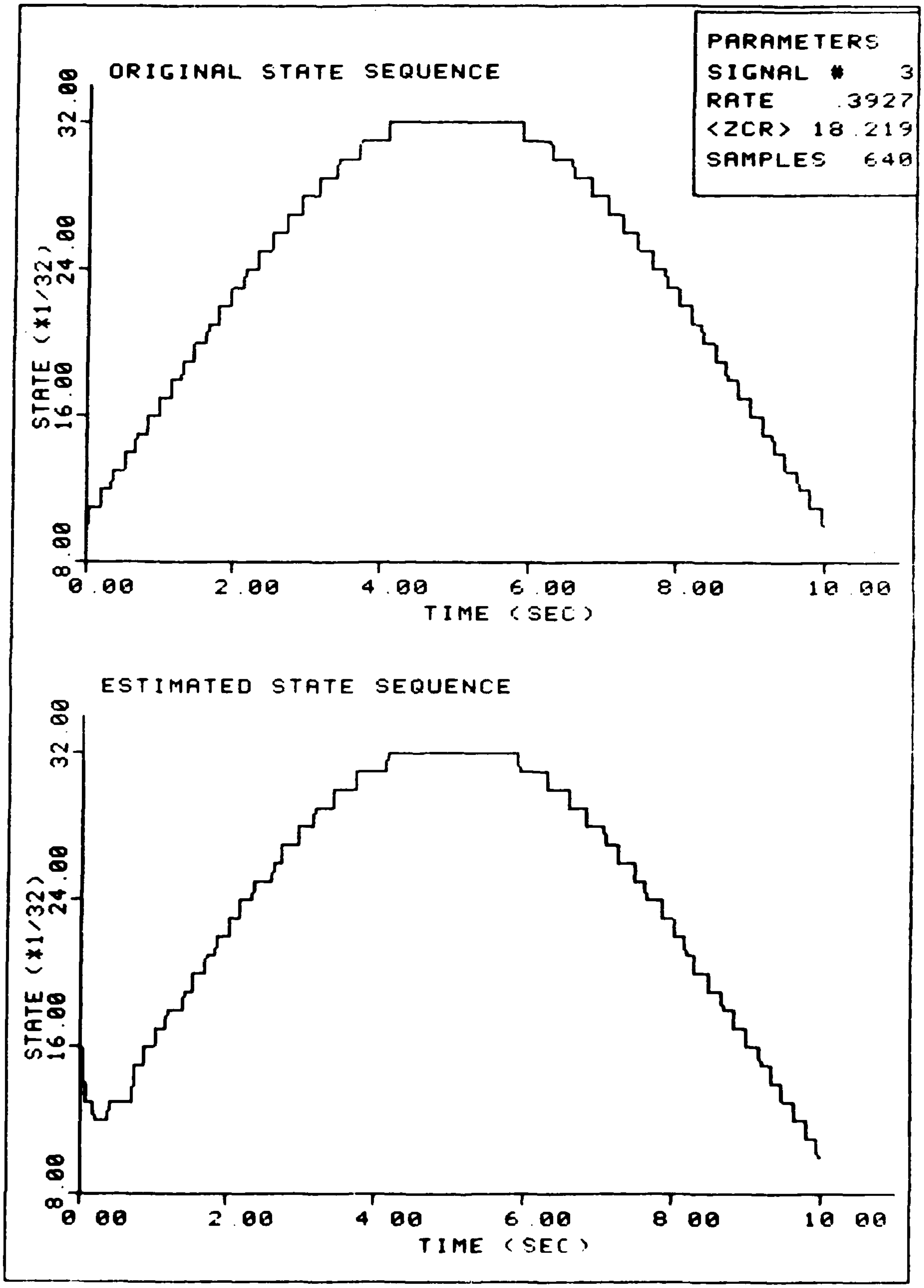


FIGURE 84. Sample Estimation

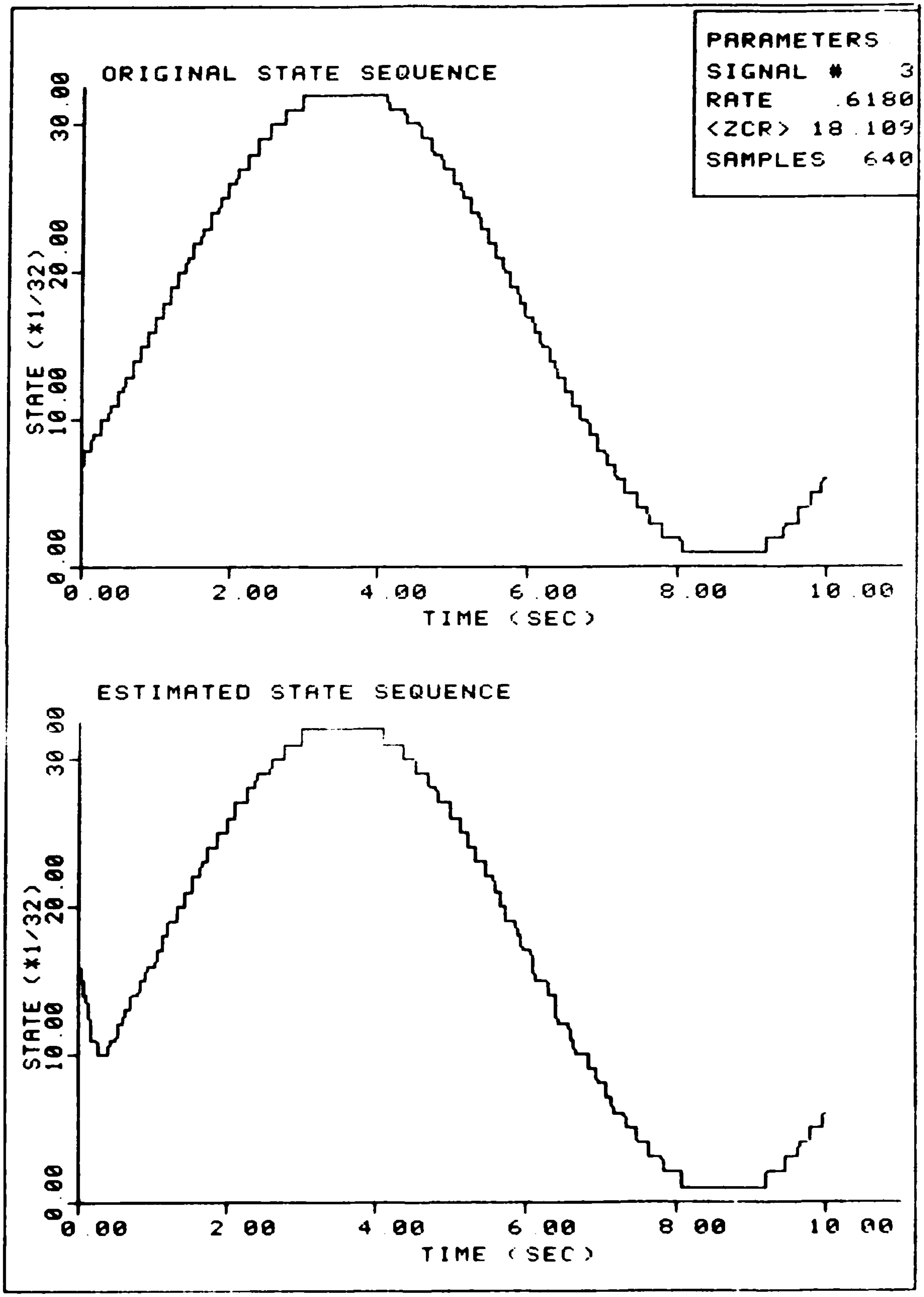


FIGURE 85. Sample Estimation

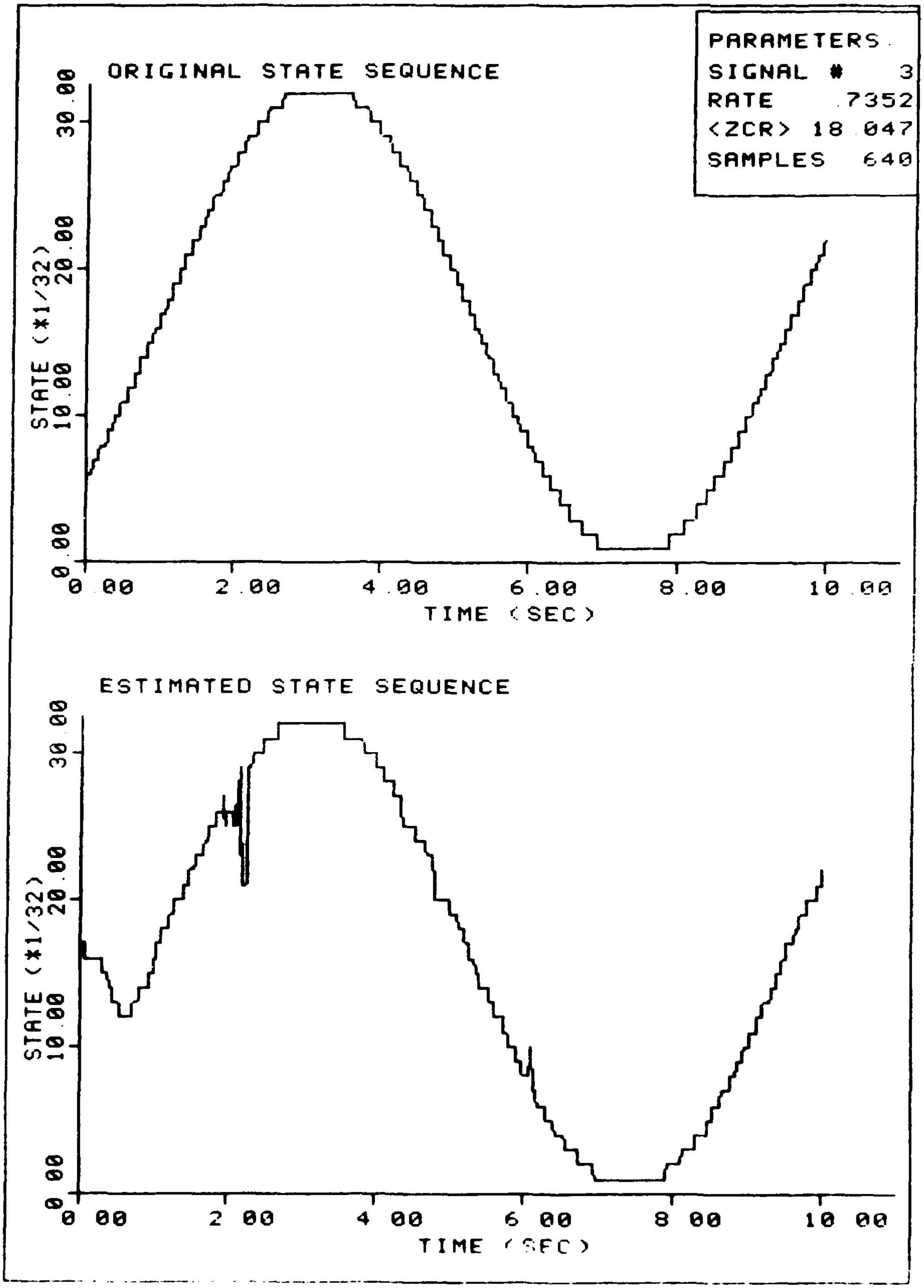


FIGURE 86. Sample Estimation

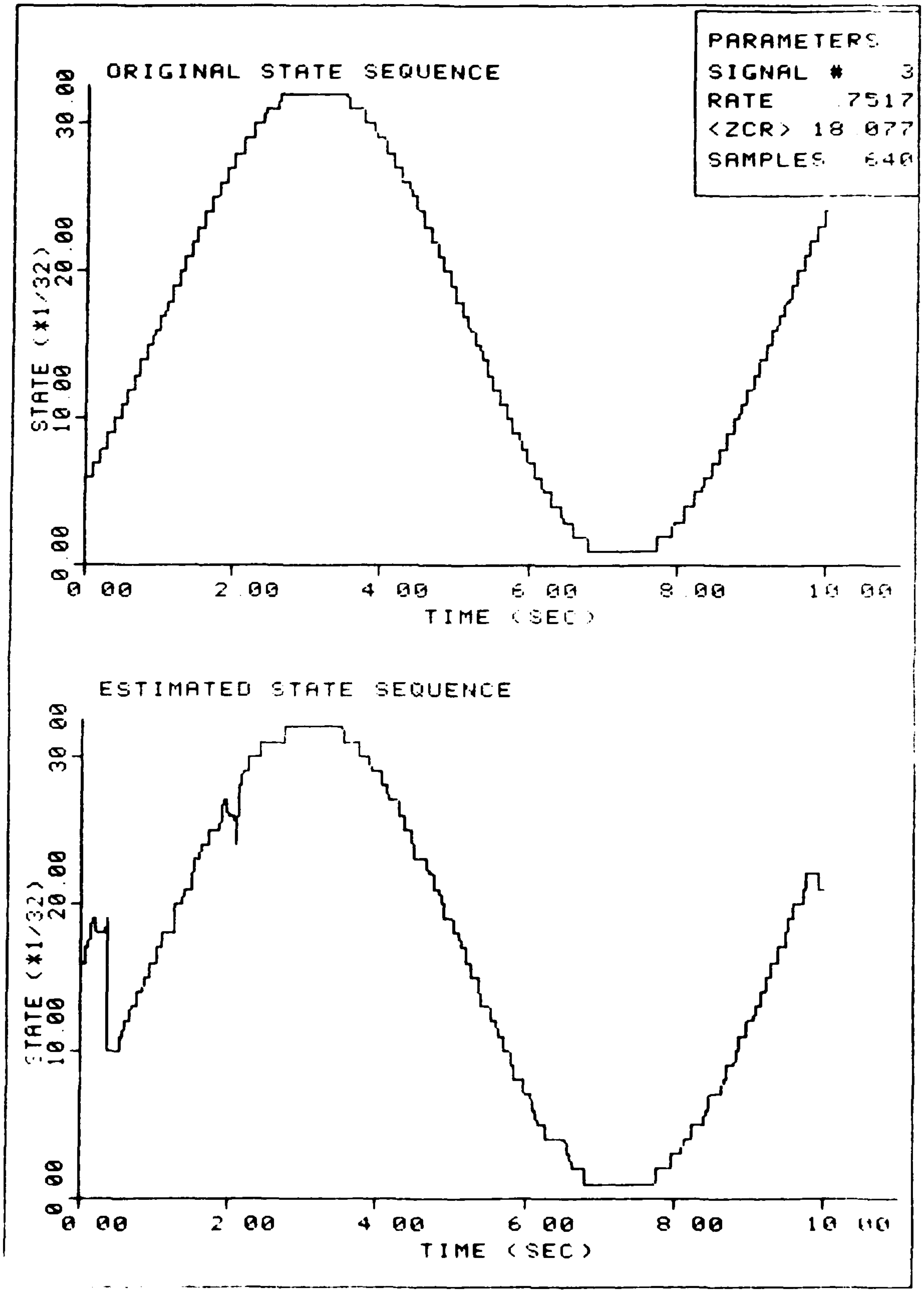


FIGURE 87. Sample Estimation

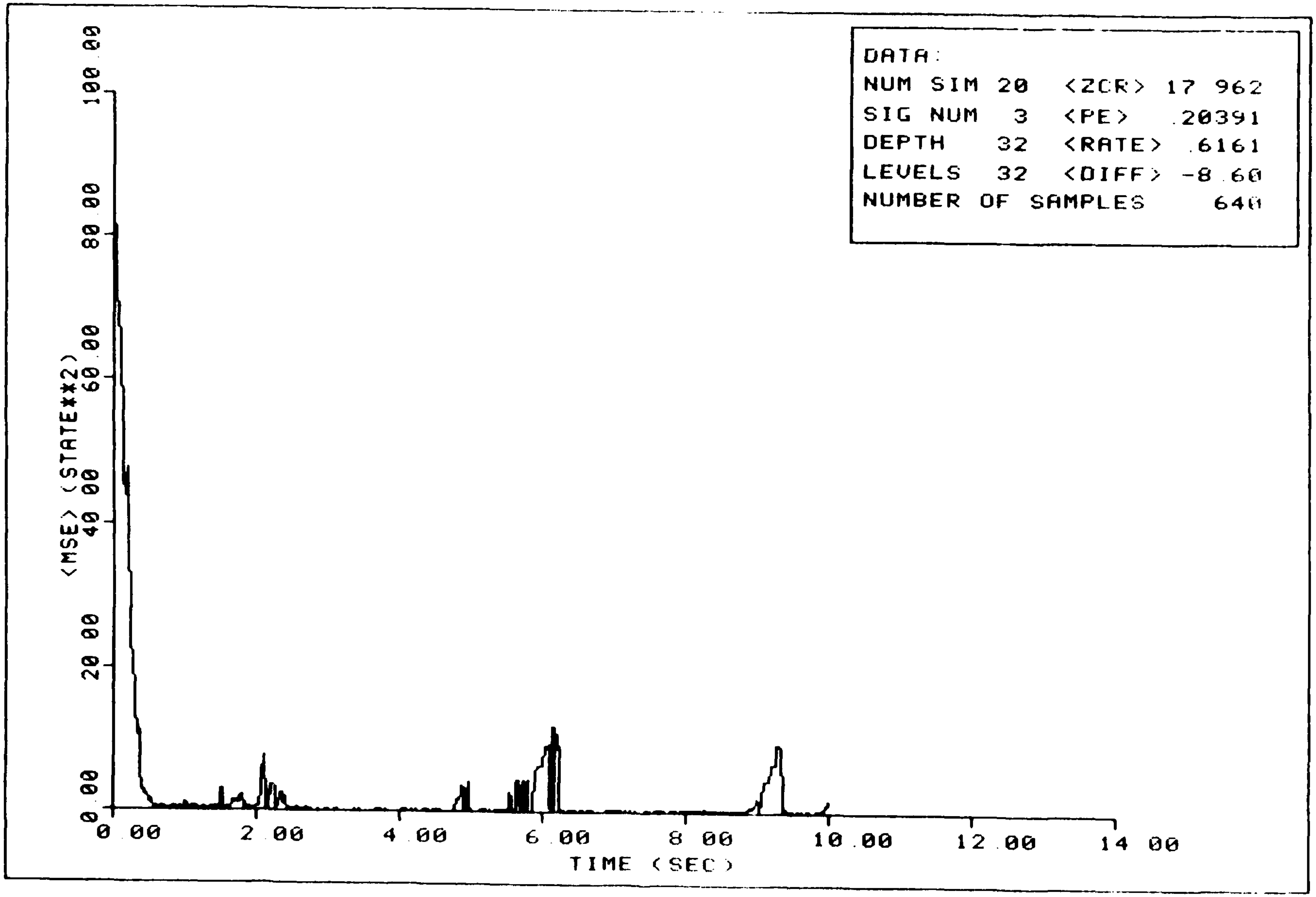


FIGURE 88. Ensemble Average MSE Performance

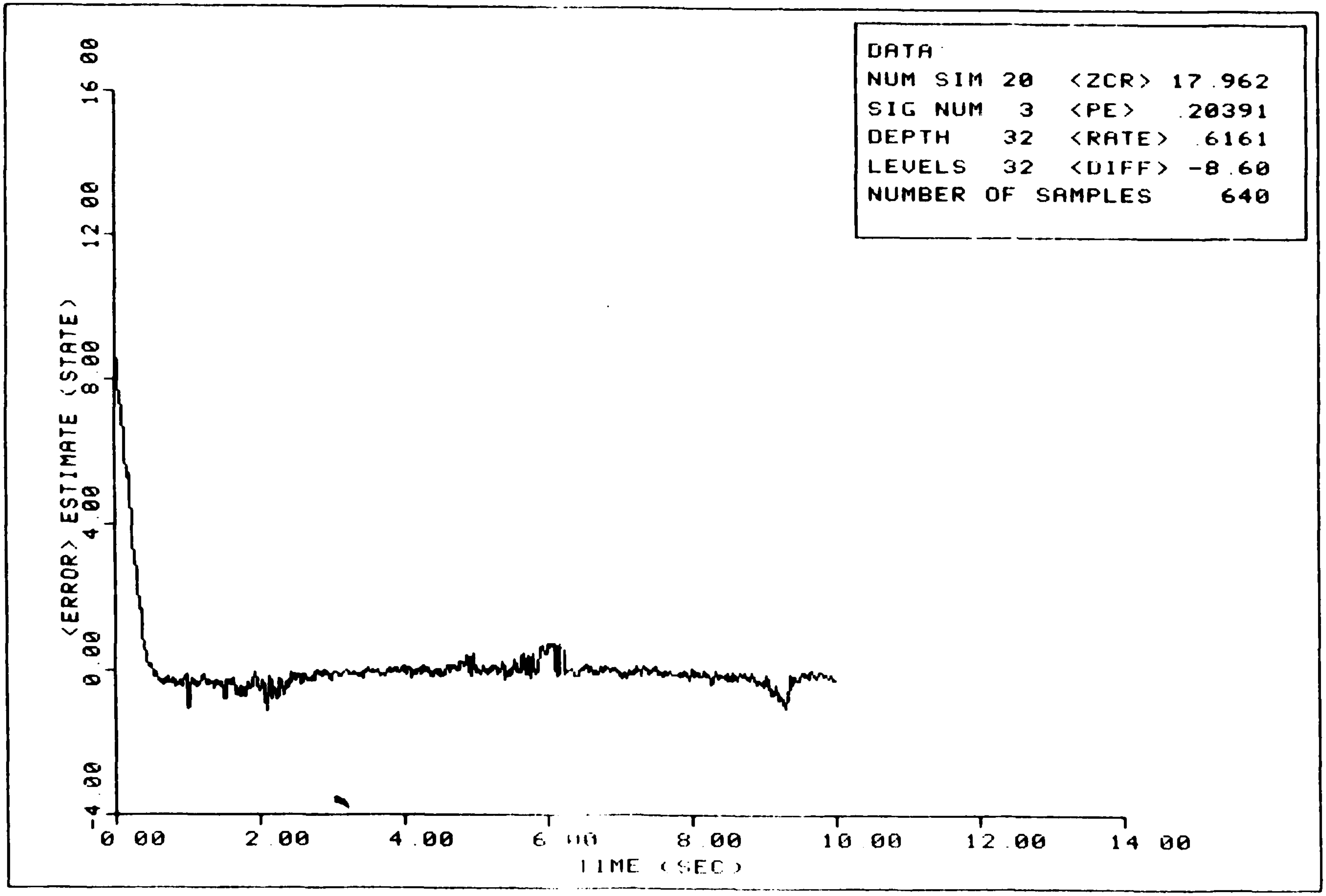


FIGURE 89. Ensemble Average State Error Performance

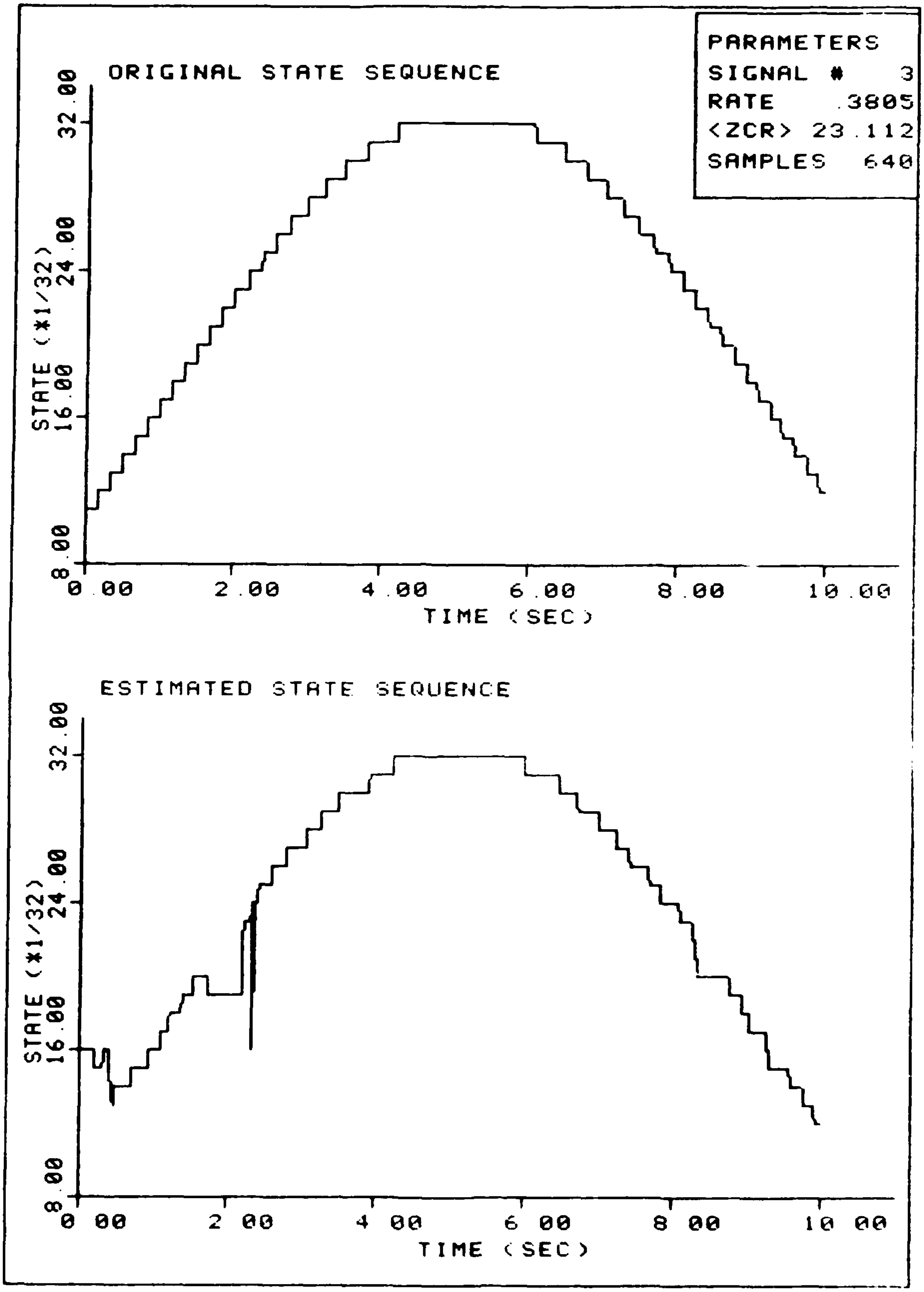


FIGURE 90. Sample Estimation

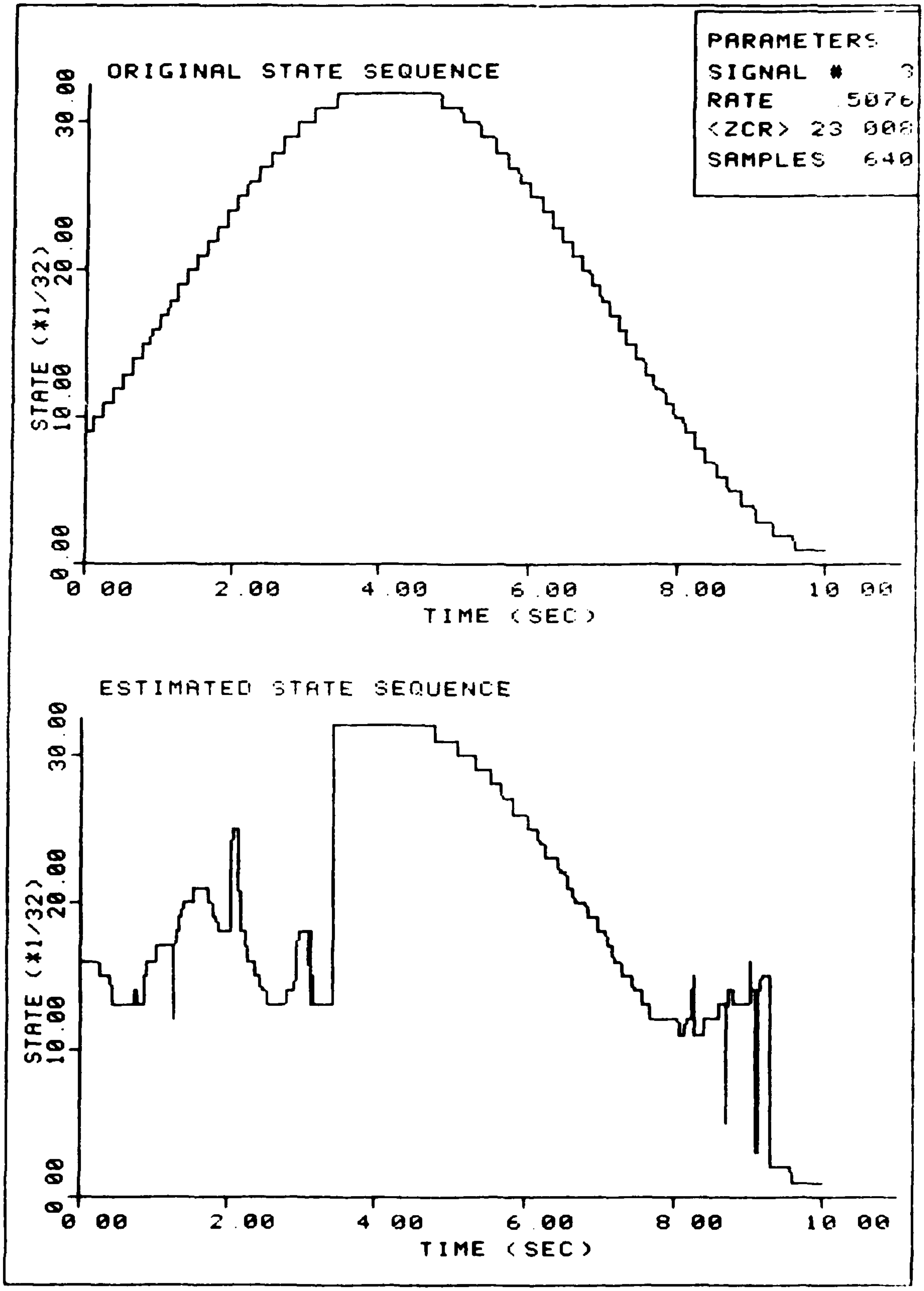


FIGURE 91. Sample Estimation

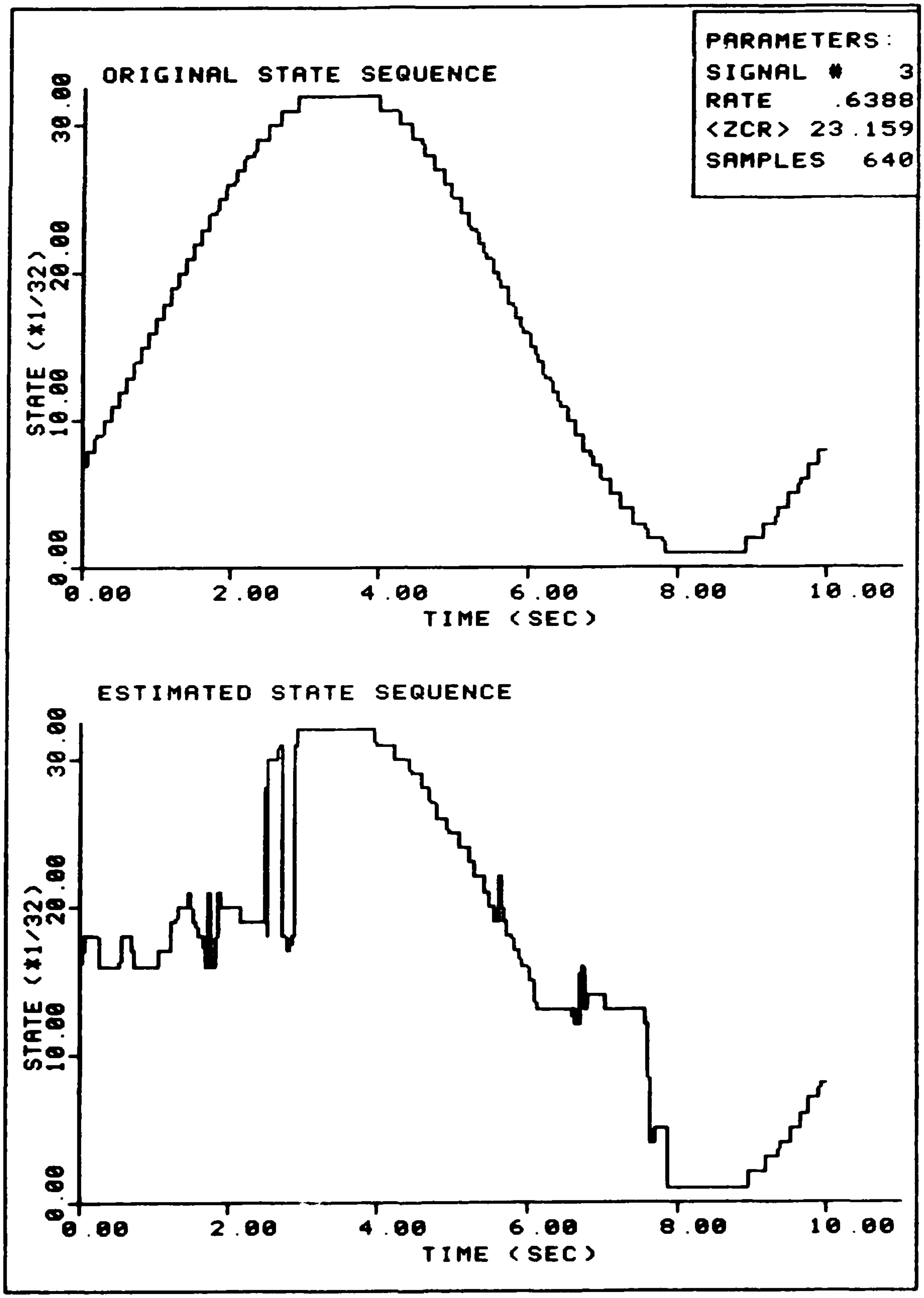


FIGURE 92. Sample Estimation

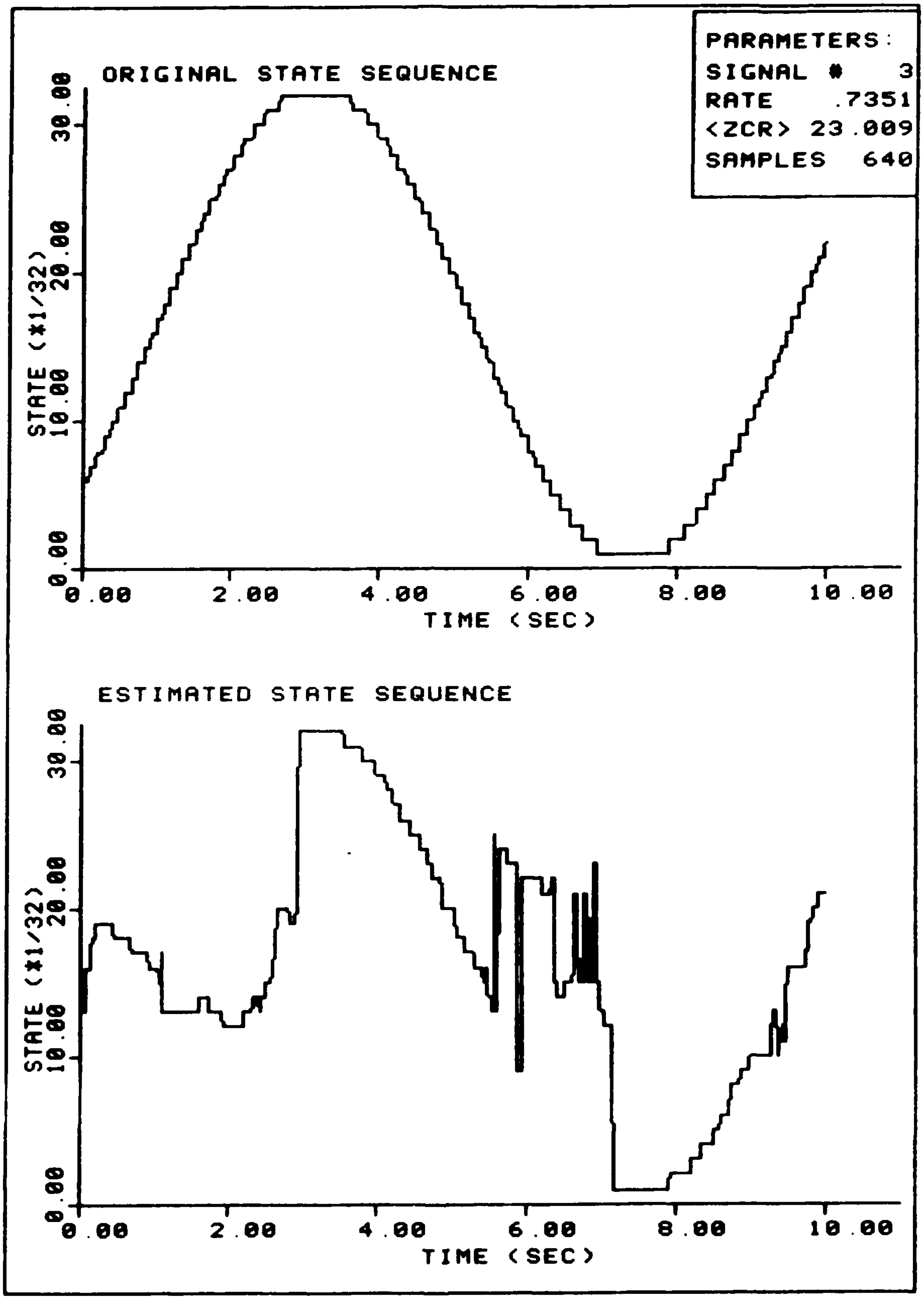


FIGURE 93. Sample Estimation

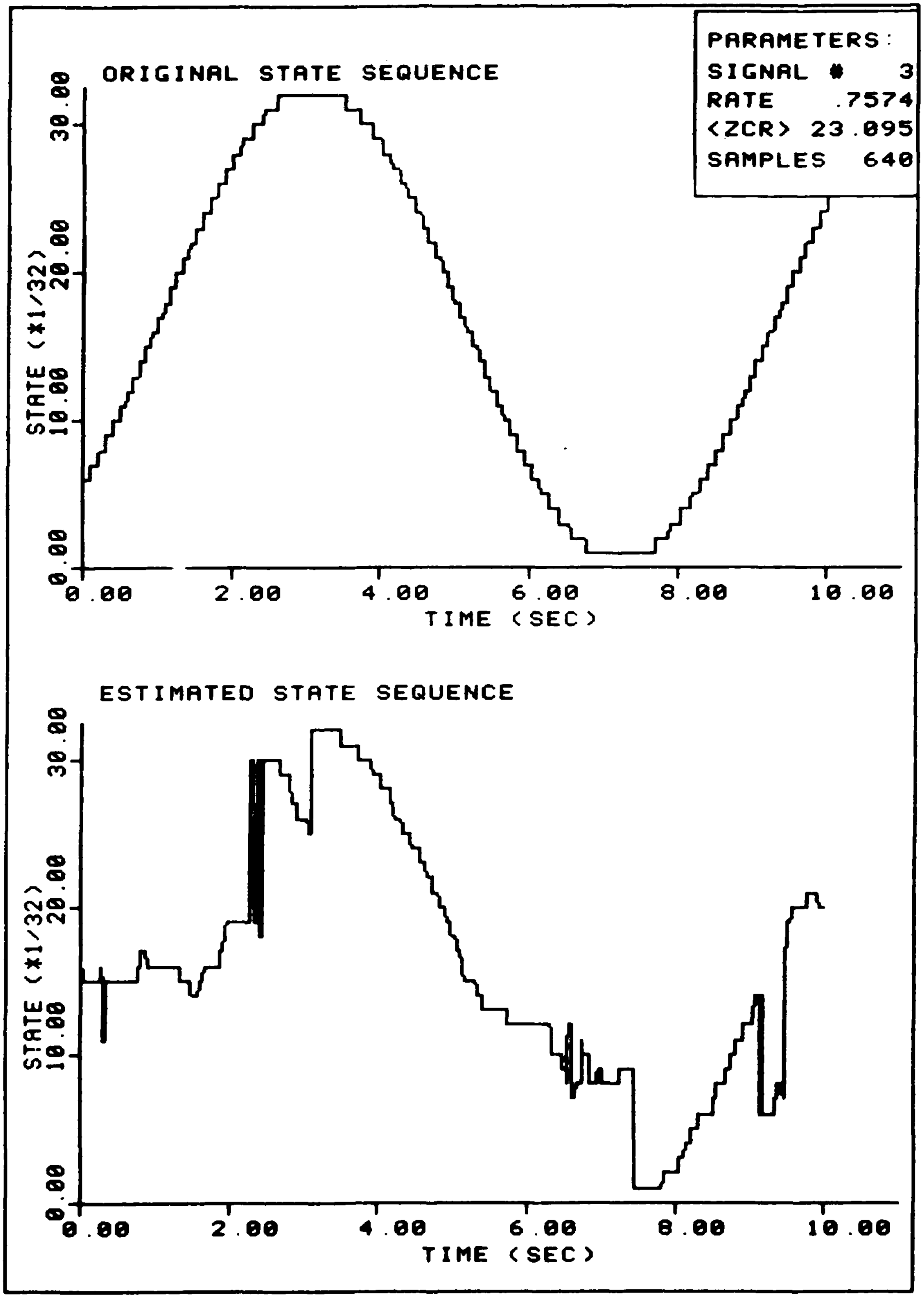


FIGURE 94. Sample Estimation

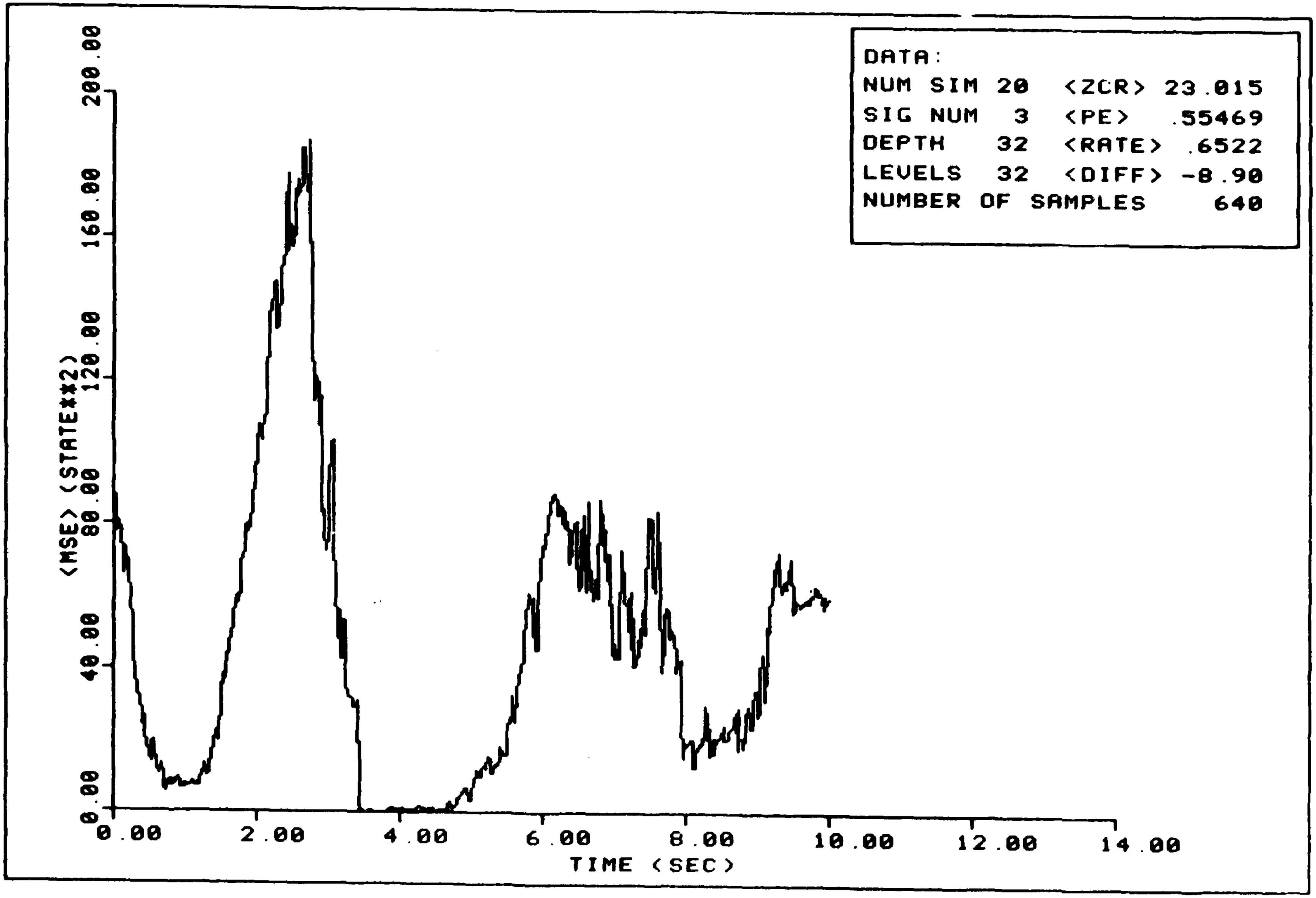


FIGURE 95. Ensemble Average MSE Performance

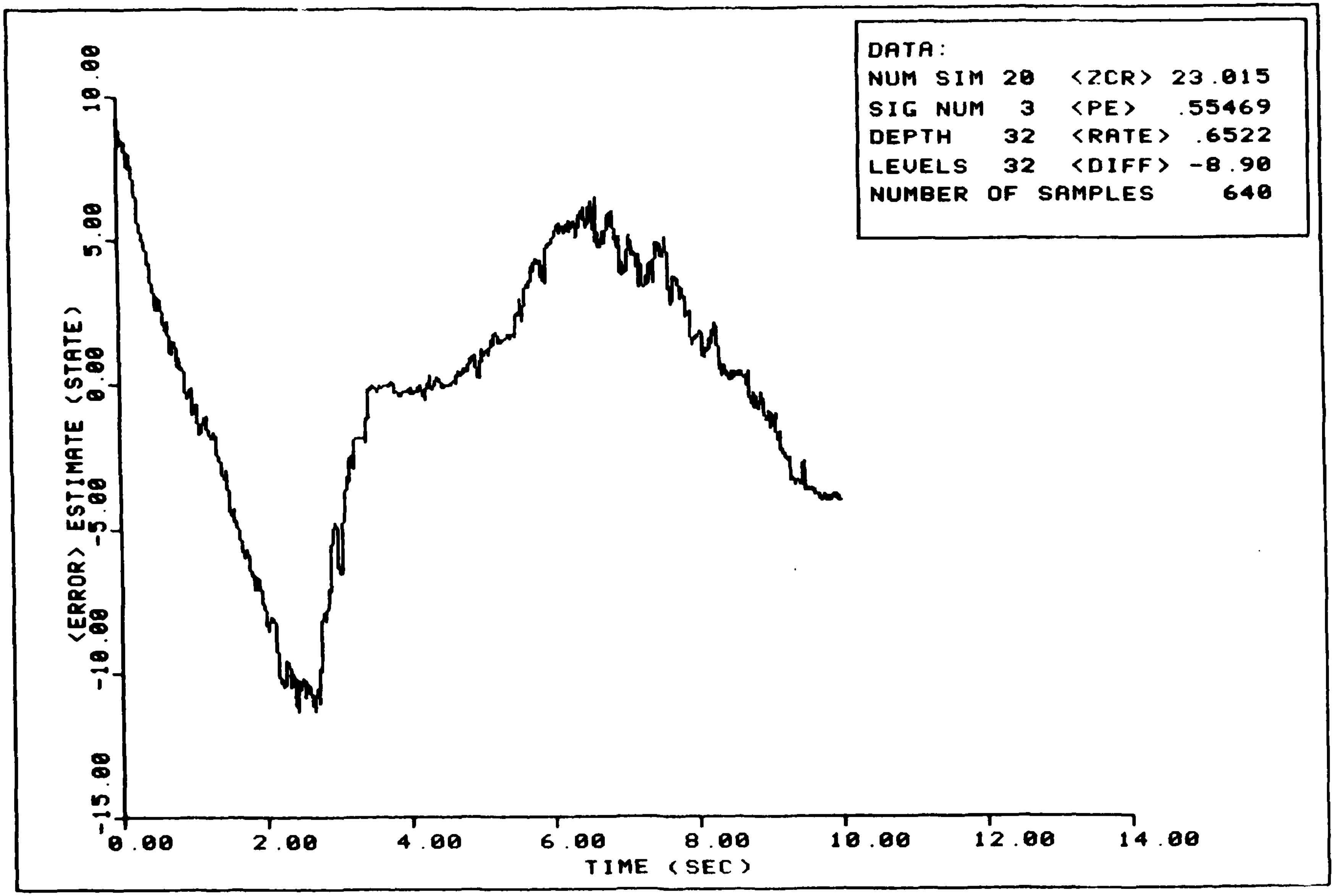


FIGURE 96. Ensemble Average State Error Performance

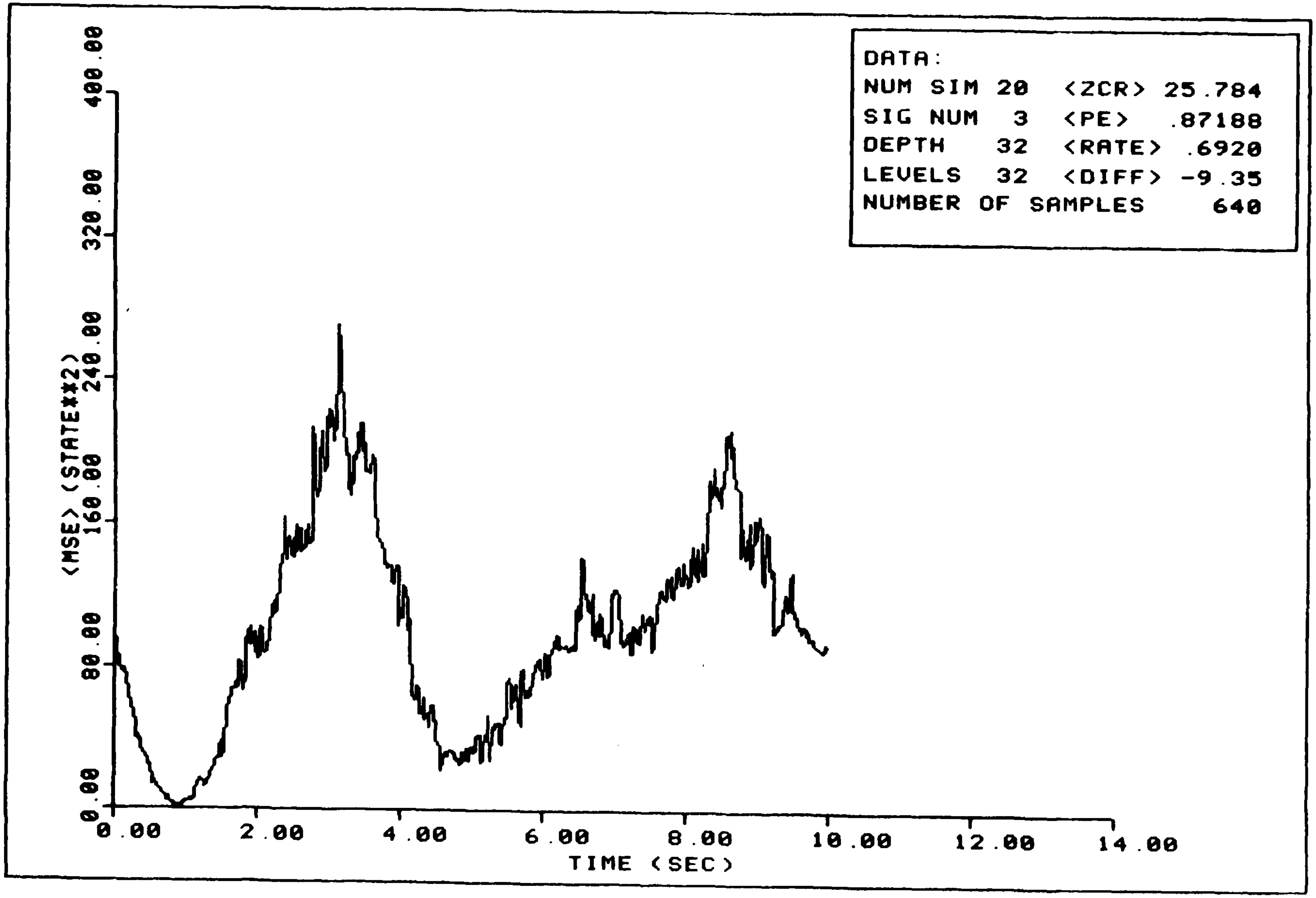


FIGURE 97. Ensemble Average MSE Performance

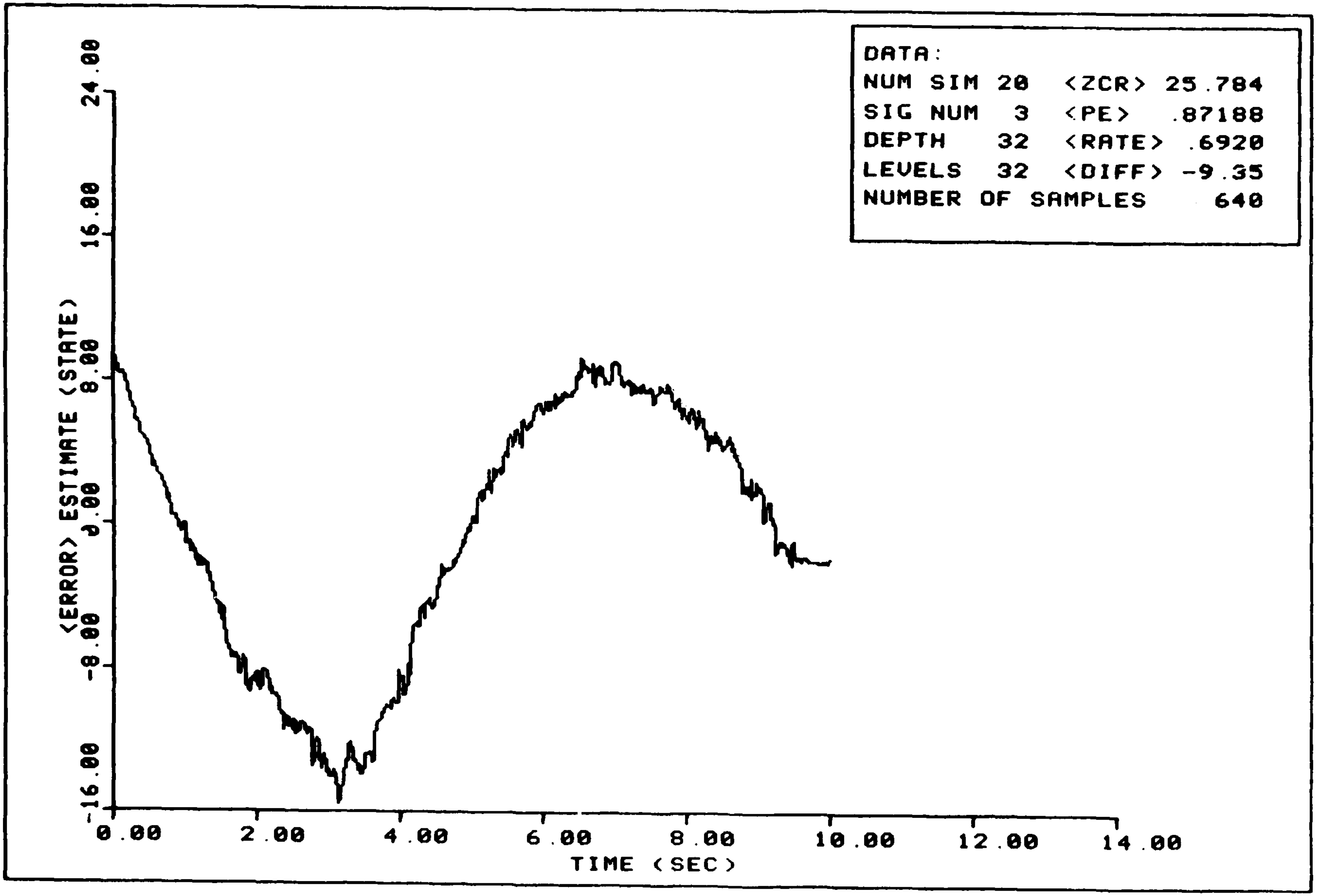


FIGURE 98. Ensemble Average State Error Performance

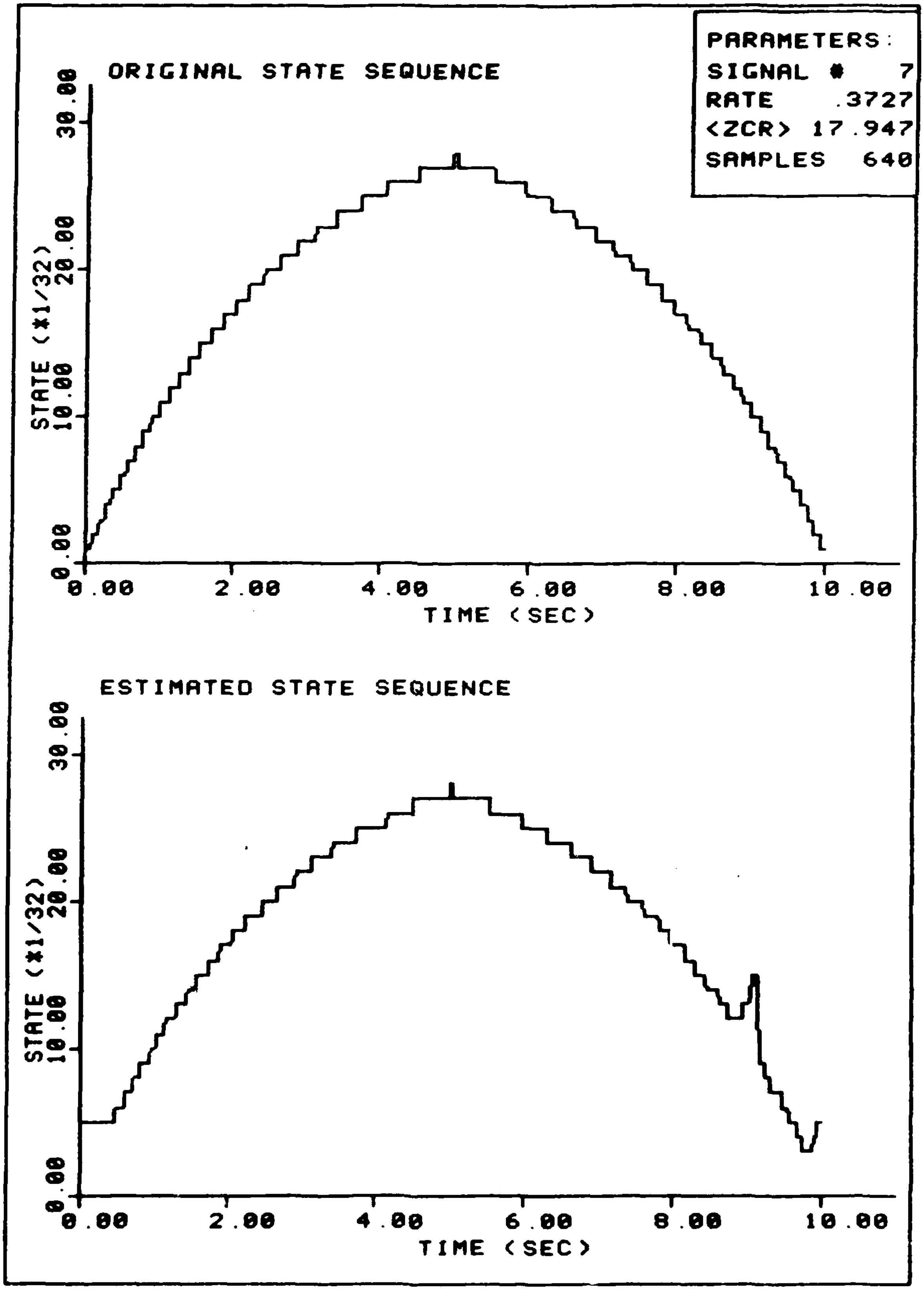


FIGURE 99. Sample Estimation

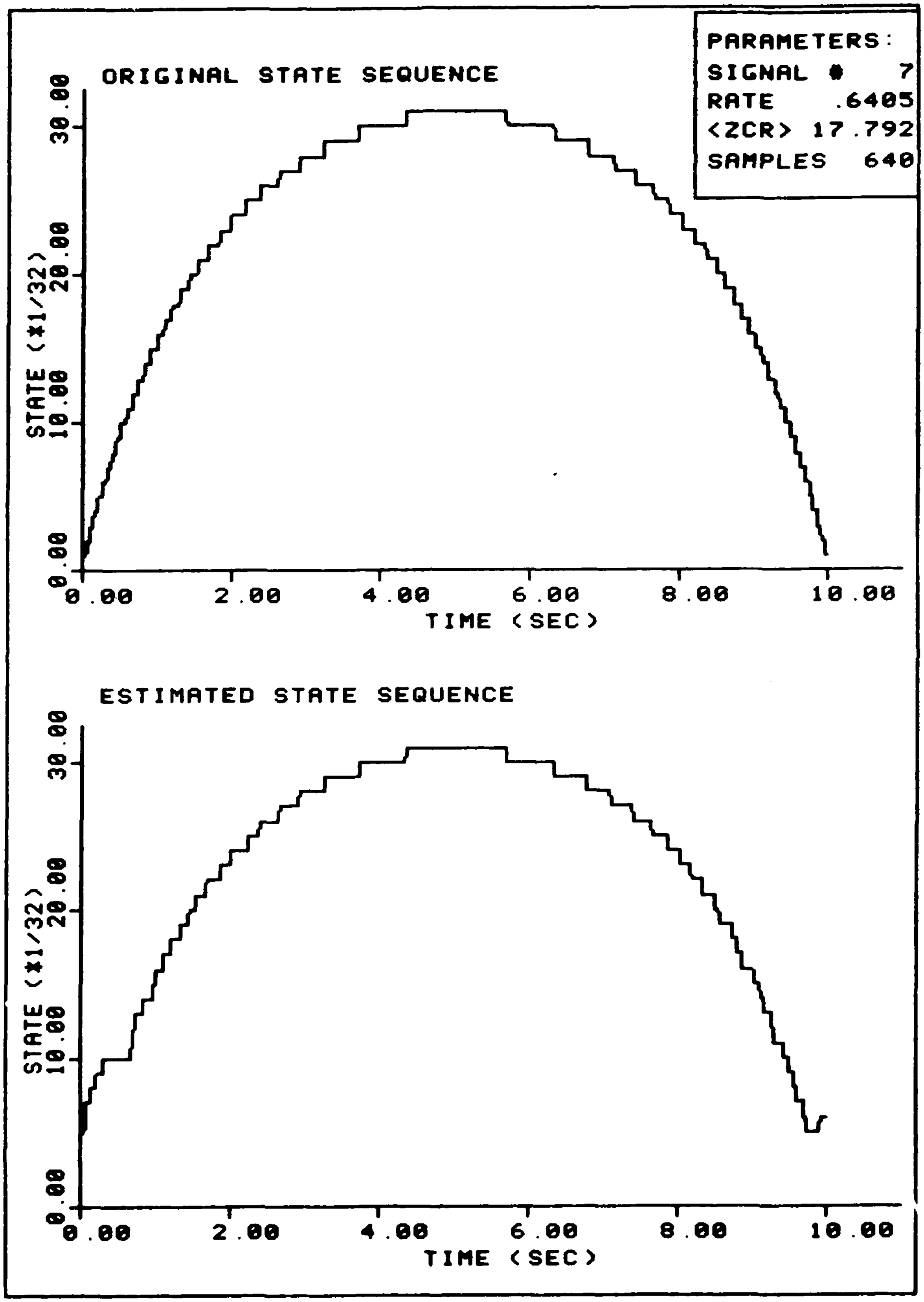


FIGURE 100. Sample Estimation

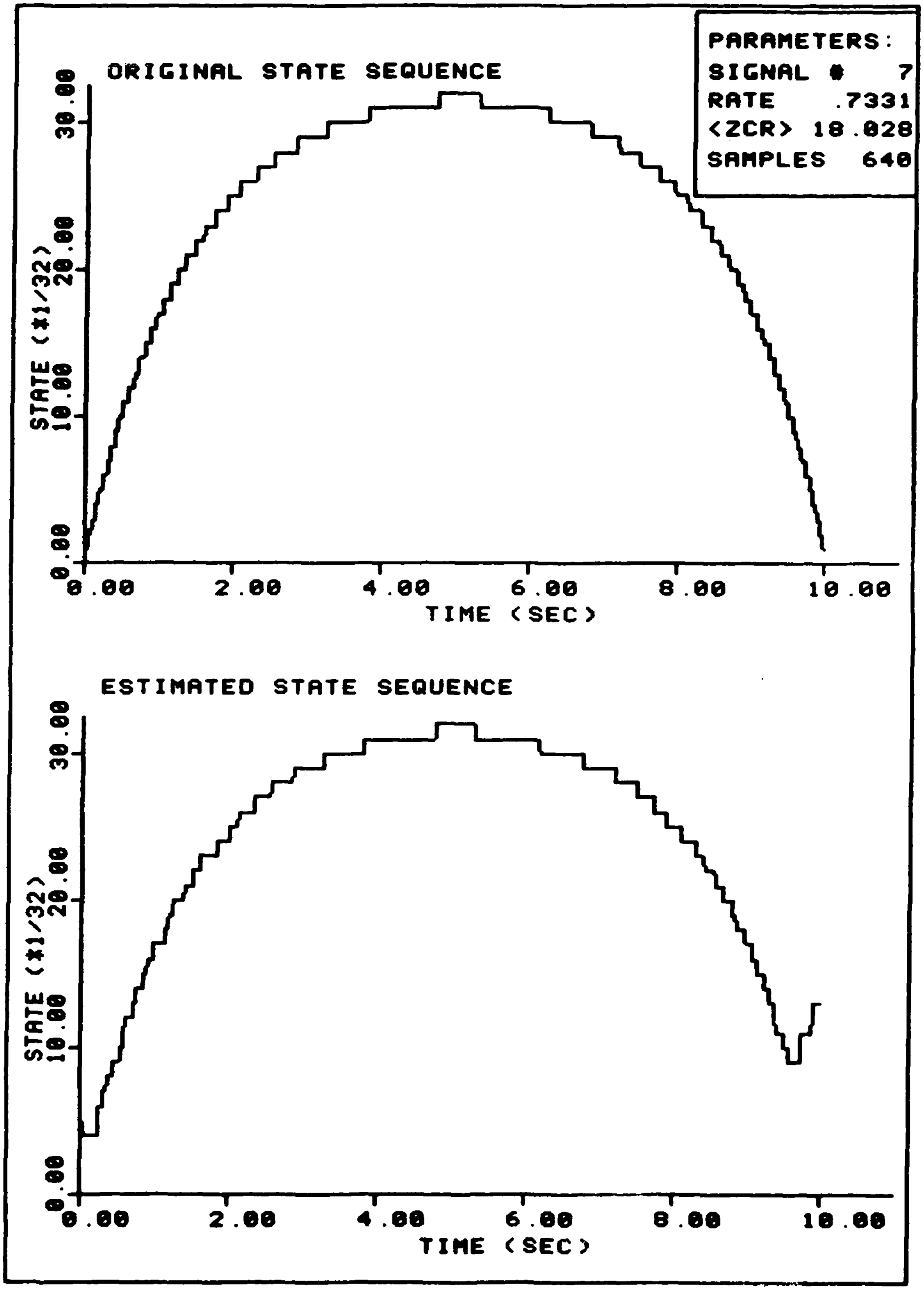


FIGURE 101. Sample Estimation

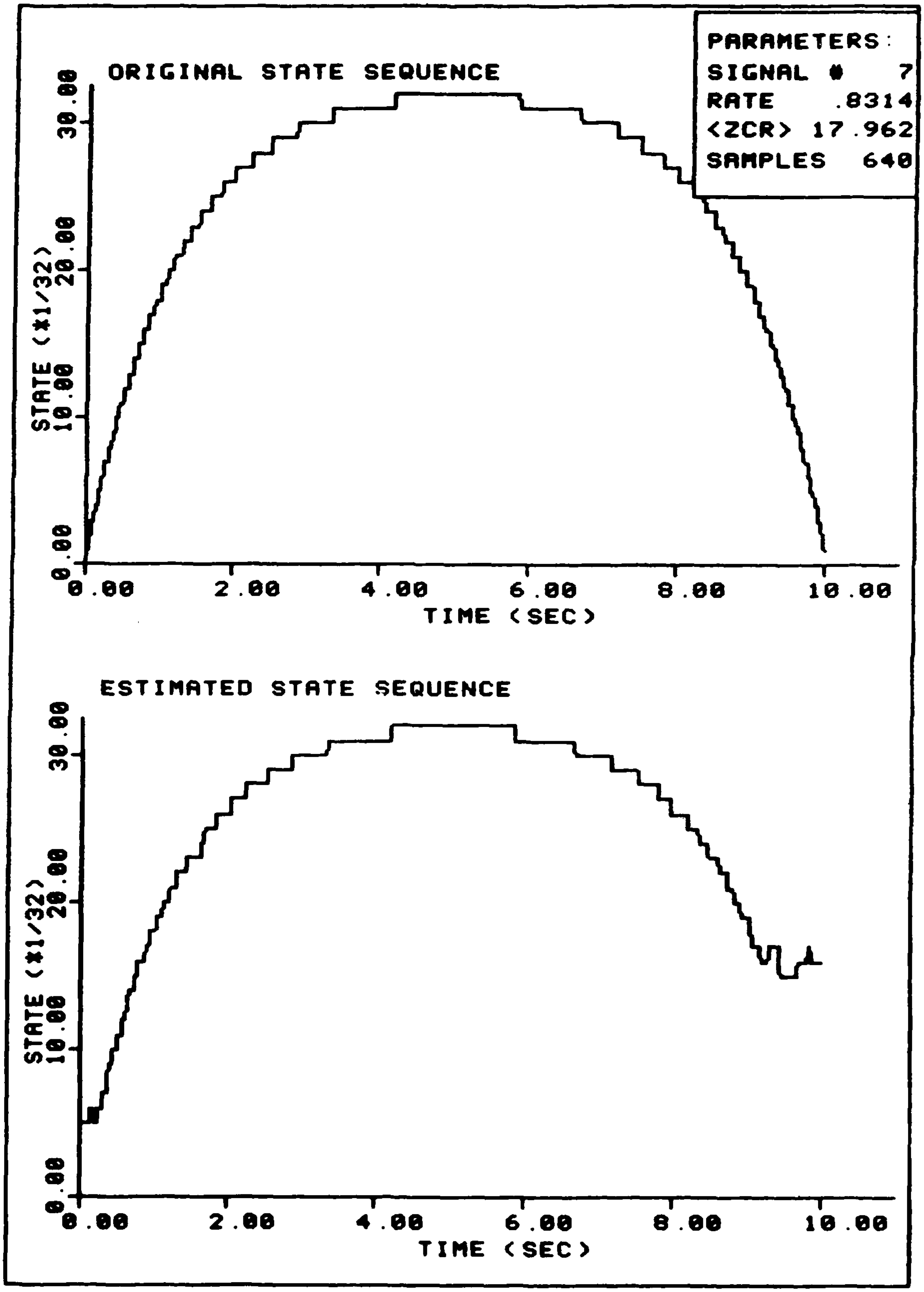


FIGURE 102. Sample Estimation

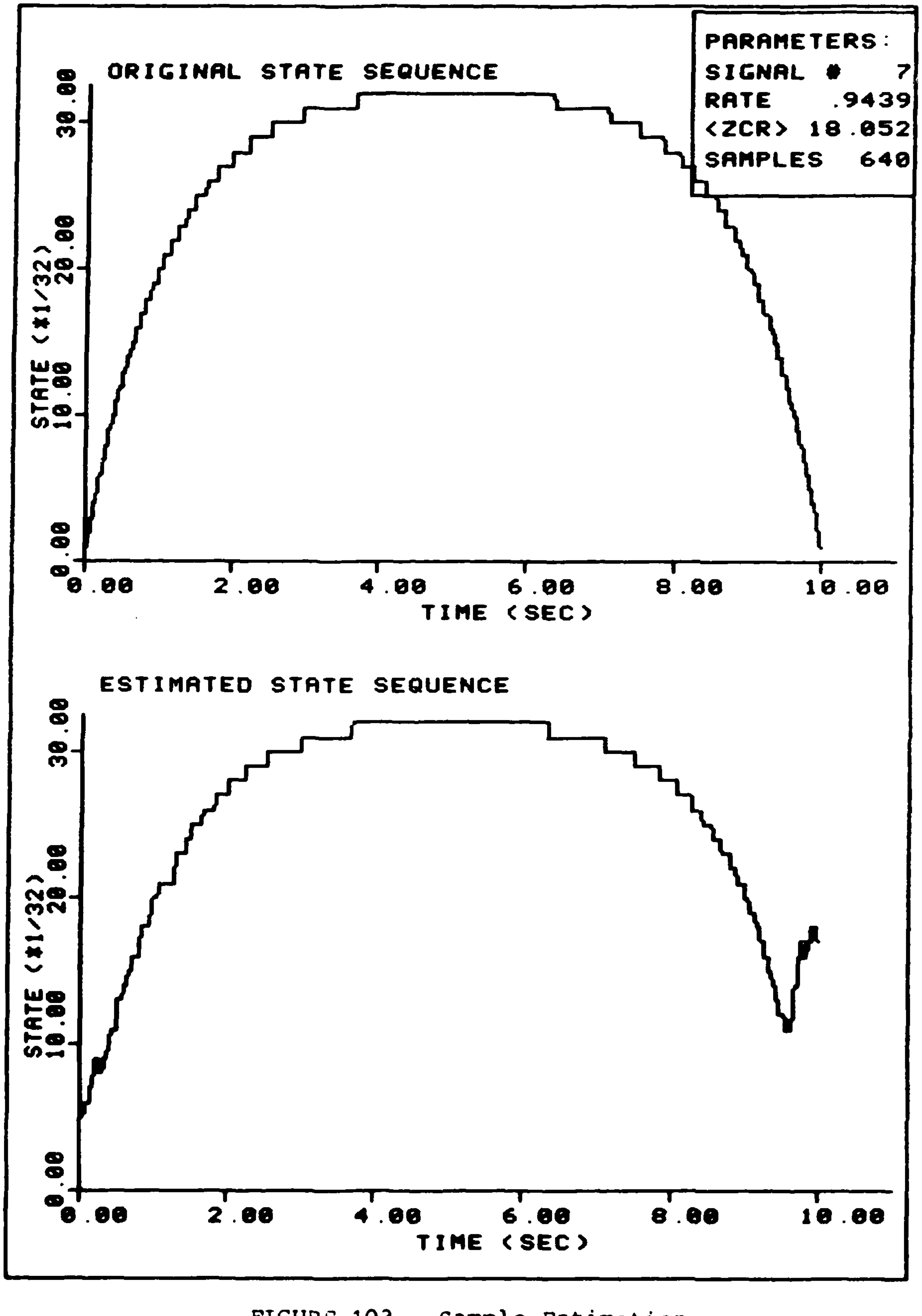


FIGURE 103. Sample Estimation

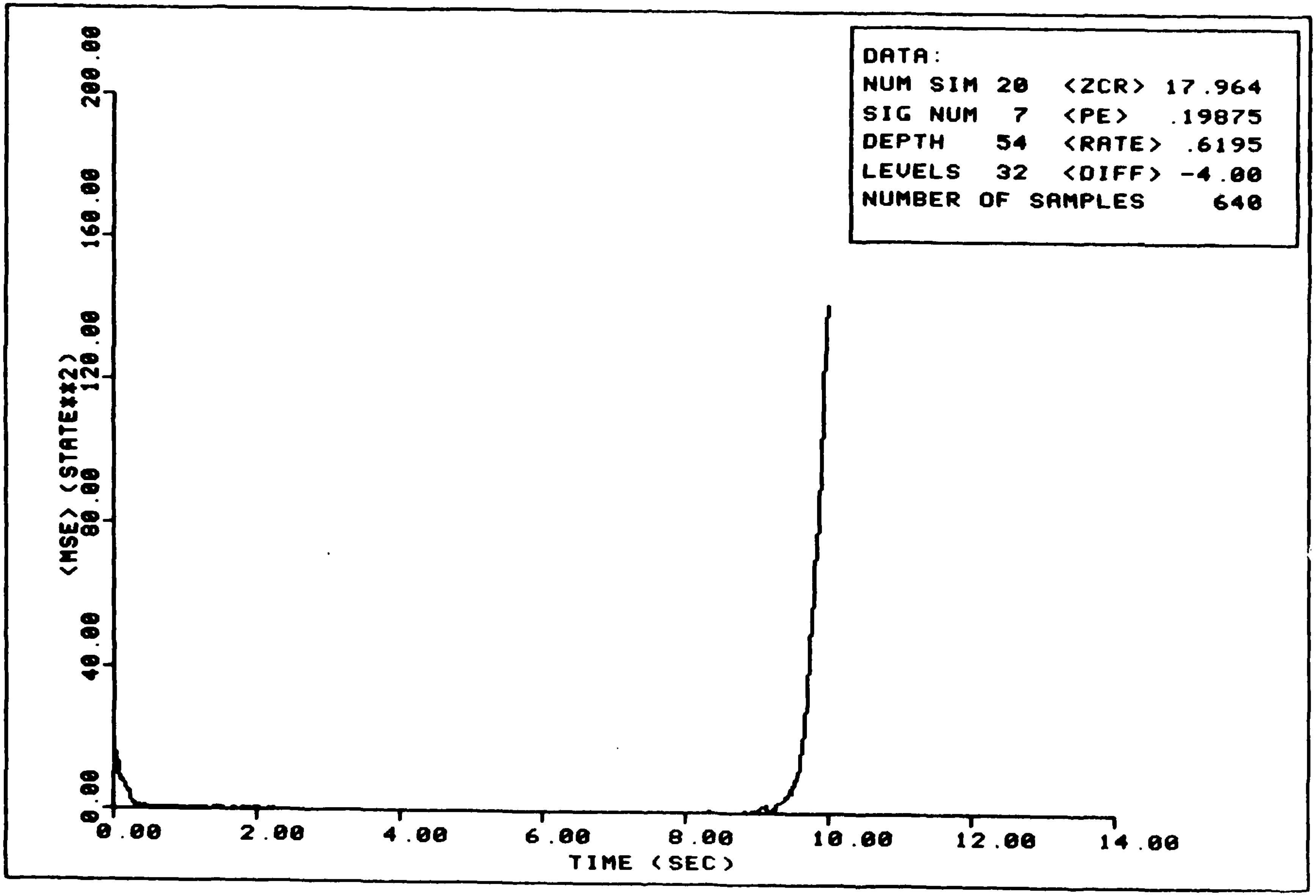


FIGURE 104. Ensemble Average MSE Performance

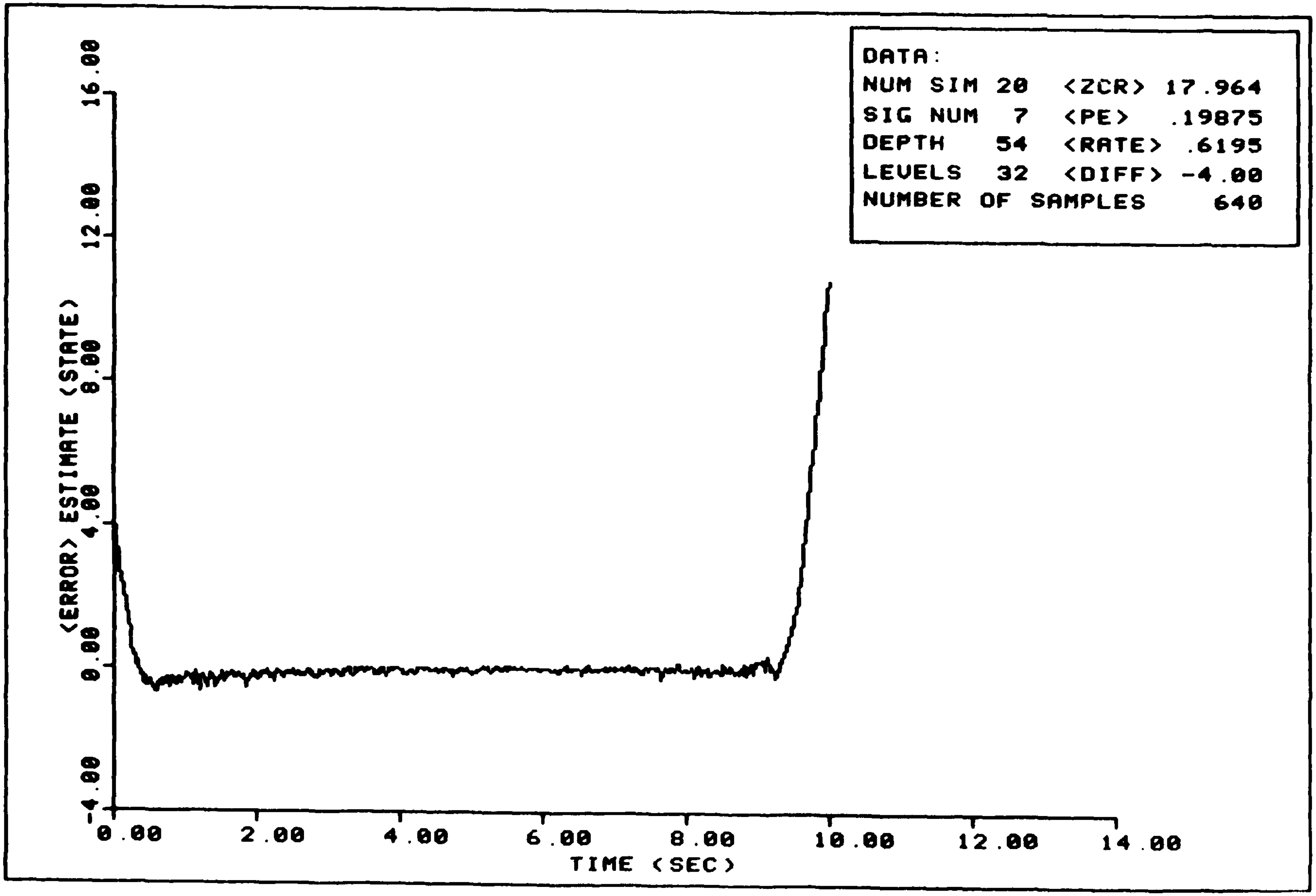


FIGURE 105. Ensemble Average State Error Performance

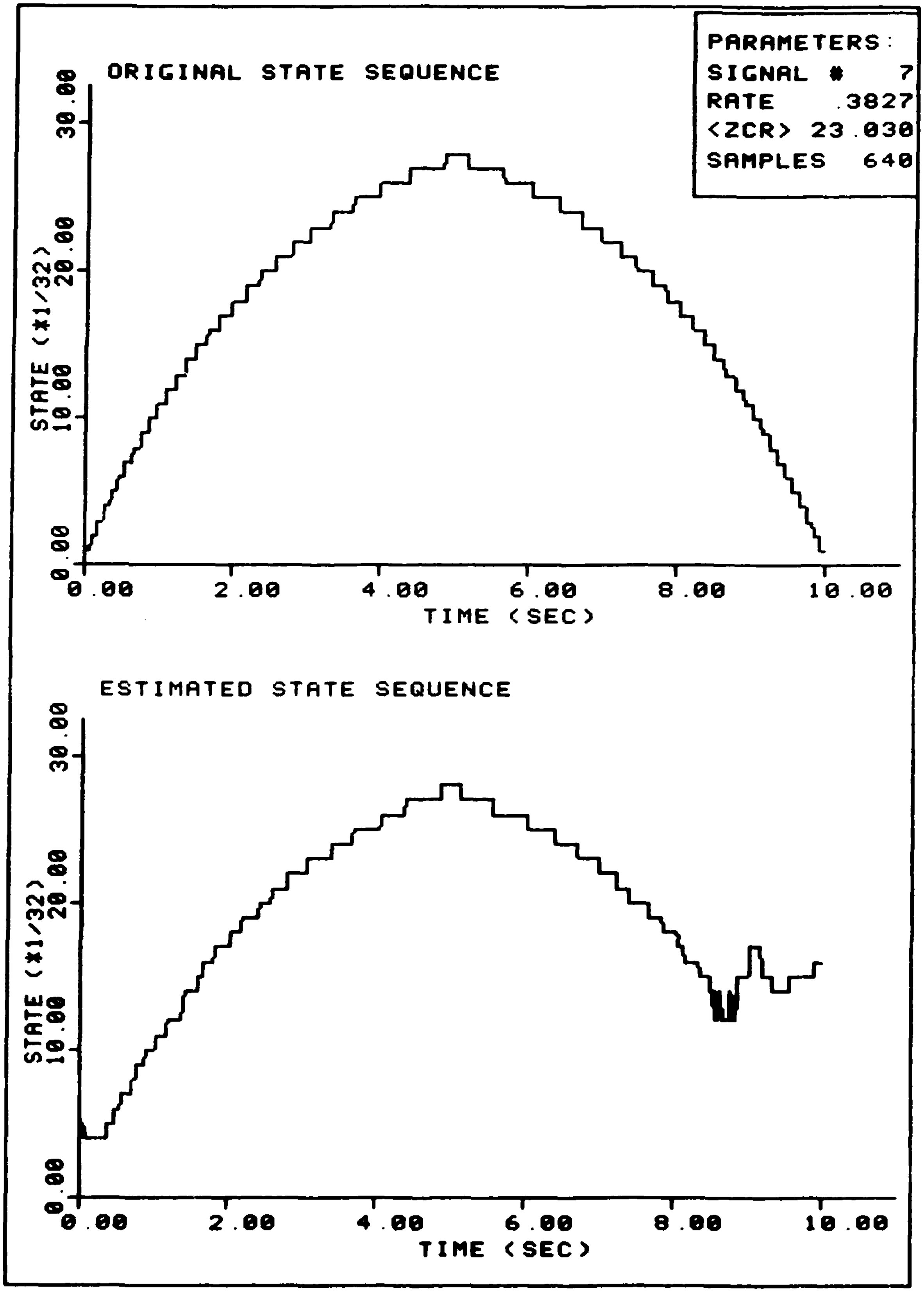


FIGURE 106. Sample Estimation

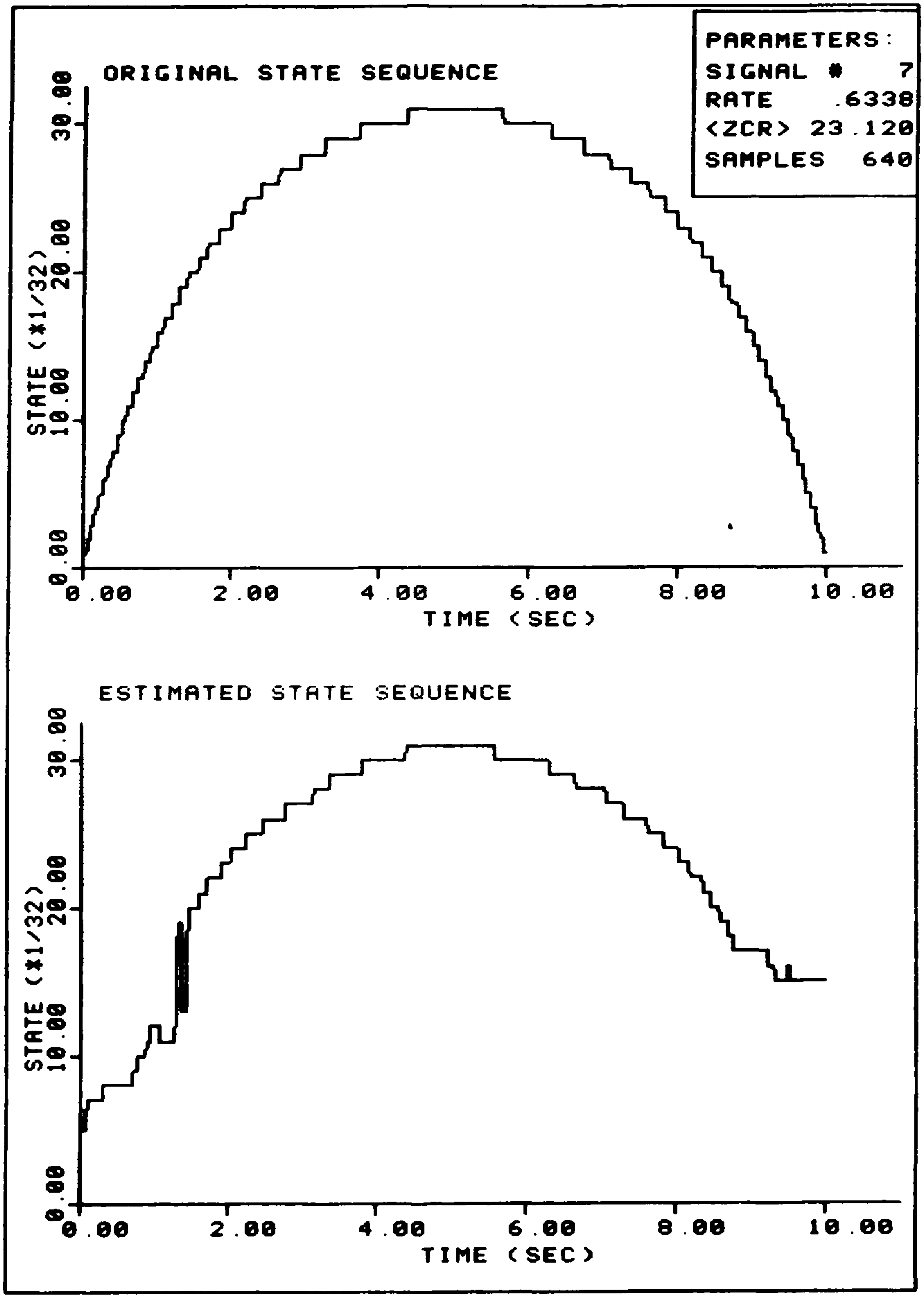


FIGURE 107. Sample Estimation

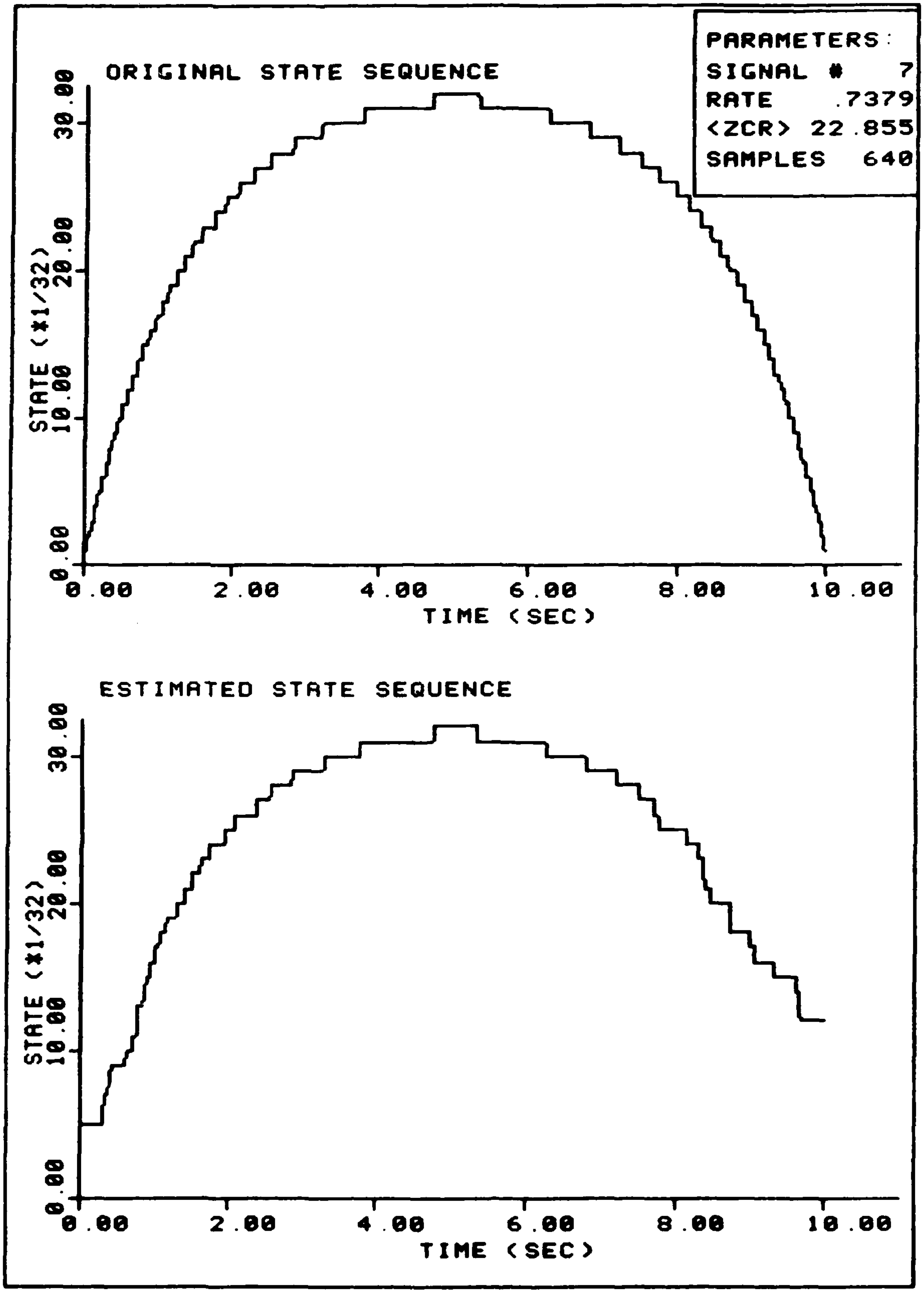


FIGURE 108. Sample Estimation

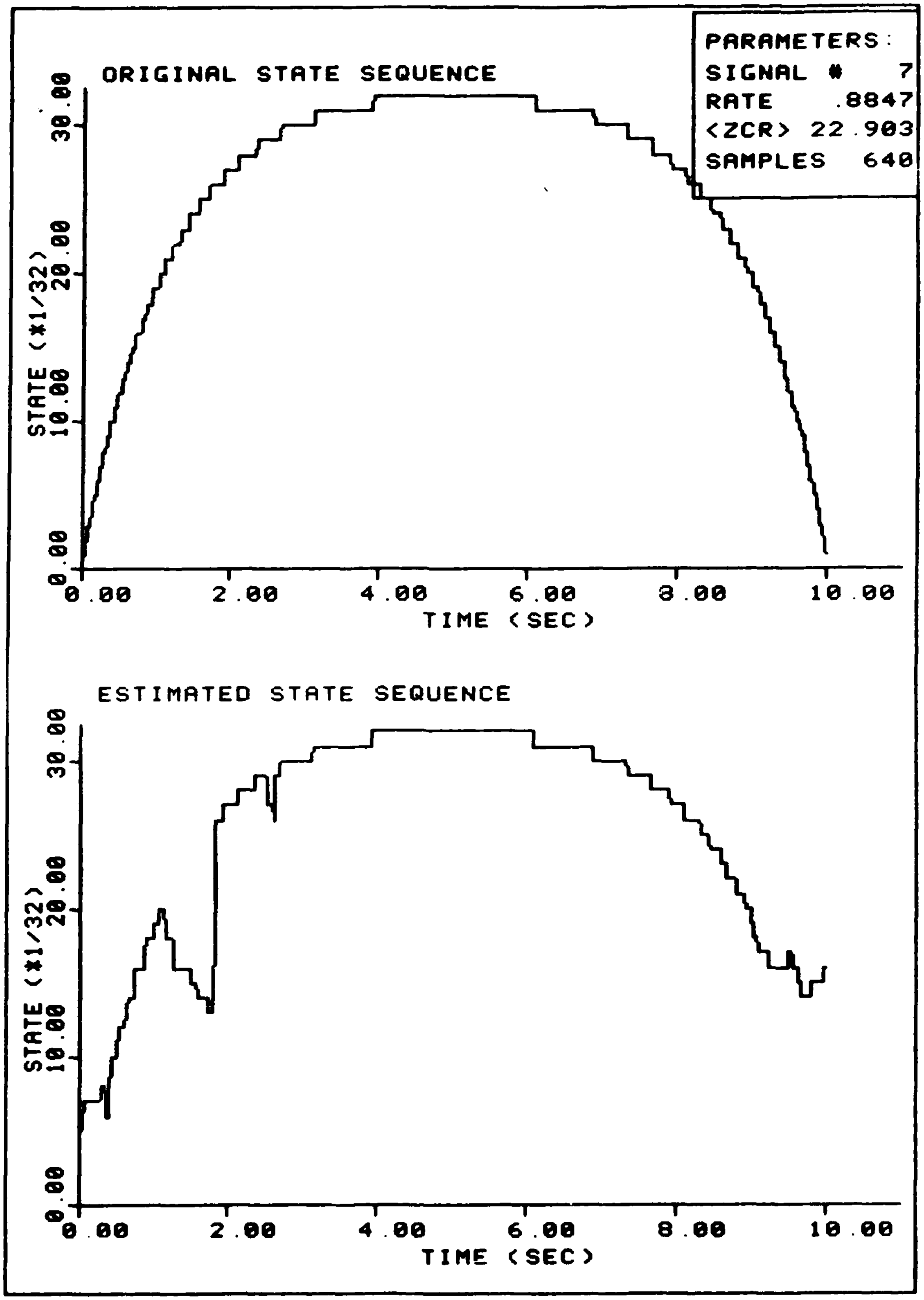
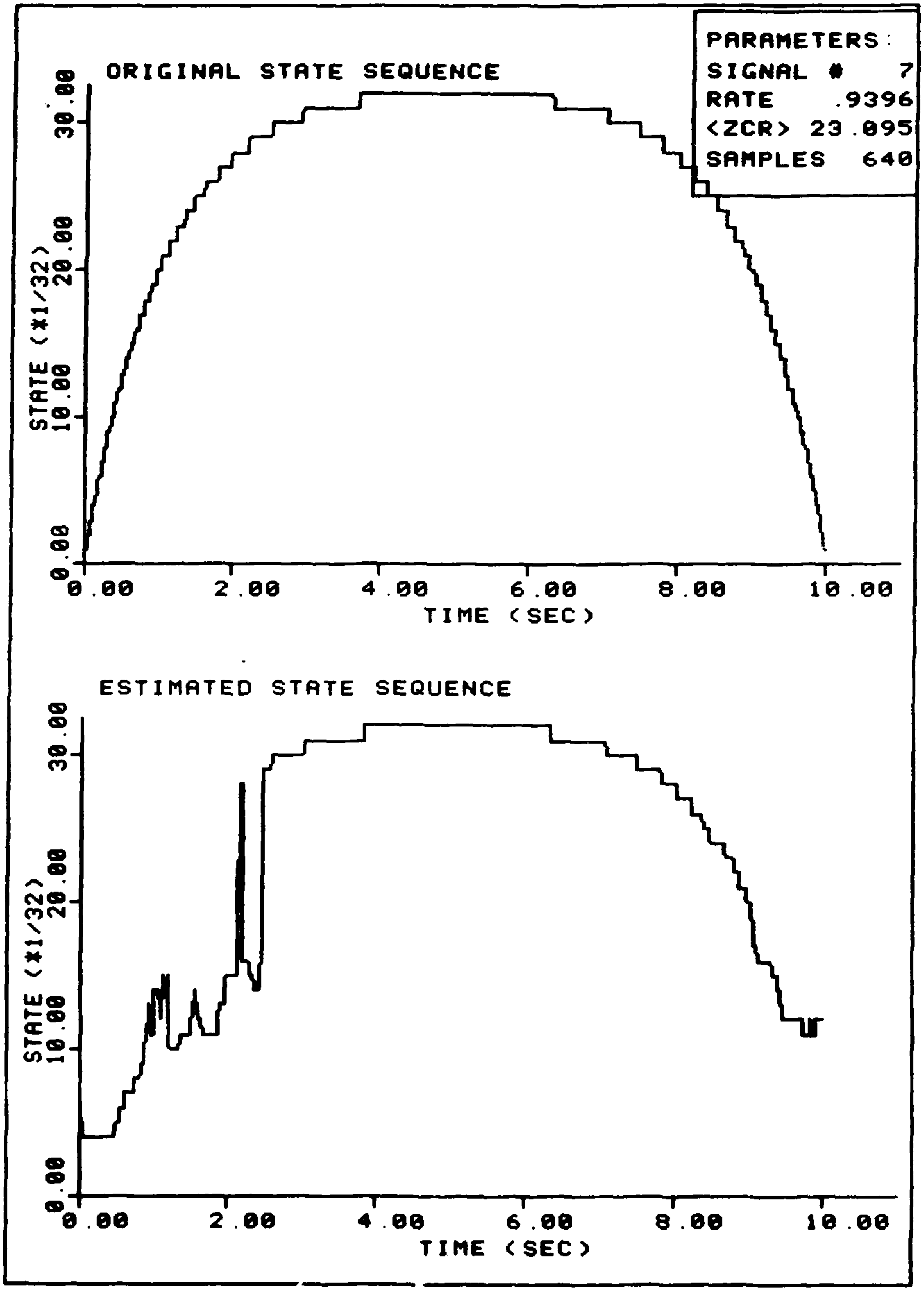


FIGURE 109. Sample Estimation



F' URE 1' . Sample Estimation

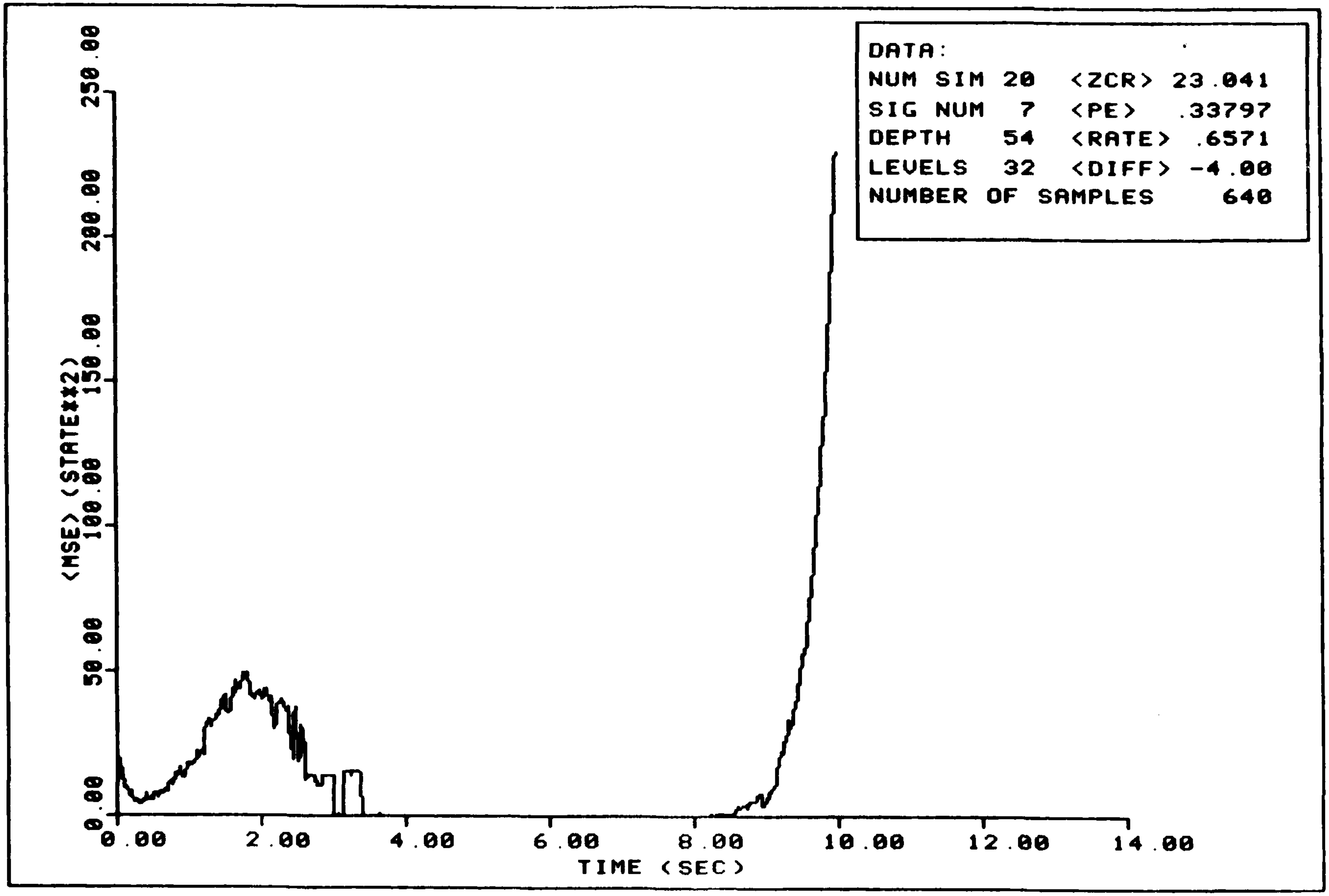


FIGURE 111. Ensemble Average MSE Performance

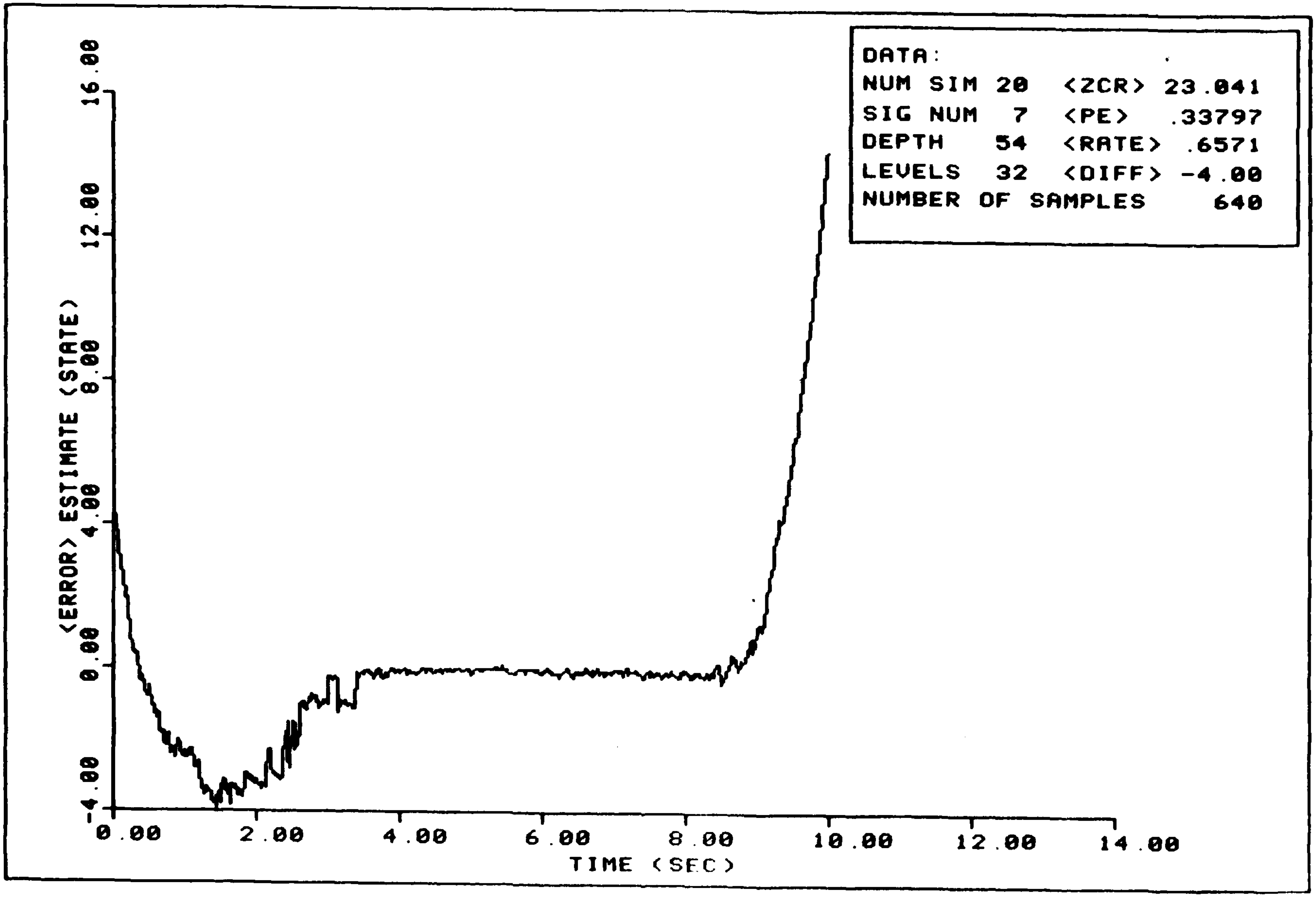


FIGURE 112. Ensemble Average State Error Performance

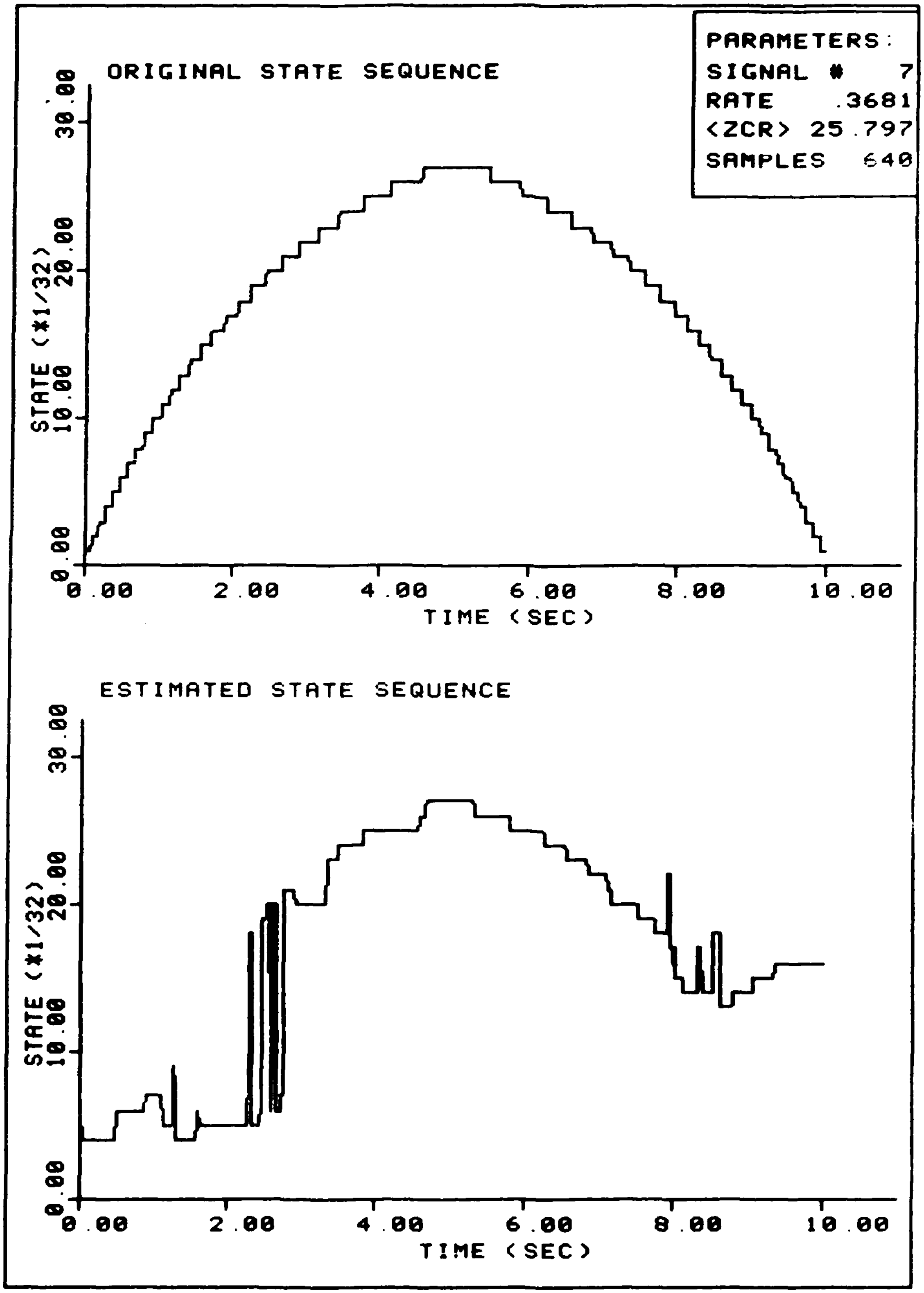


FIGURE 113. Sample Estimation

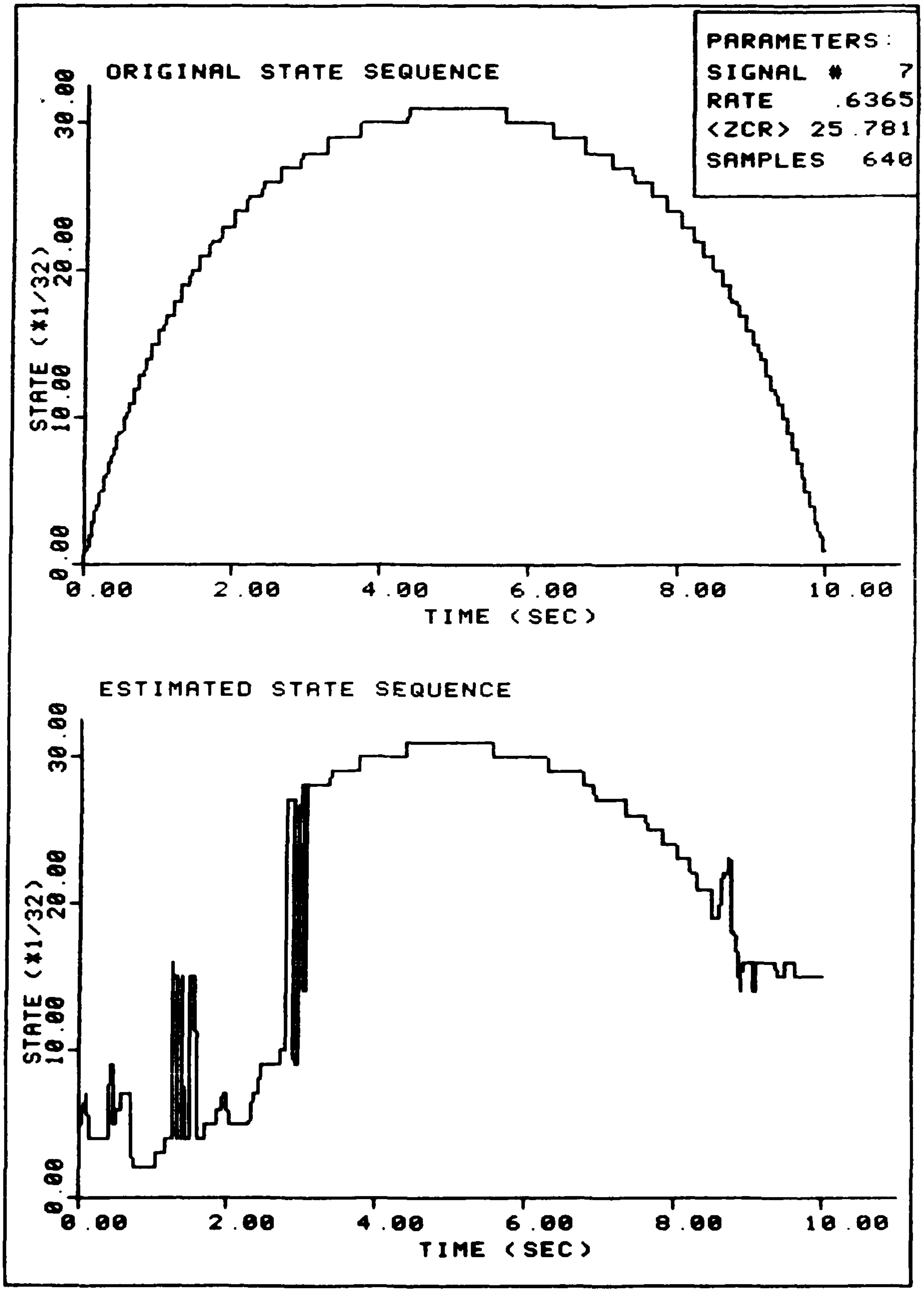


FIGURE 114. Sample Estimation

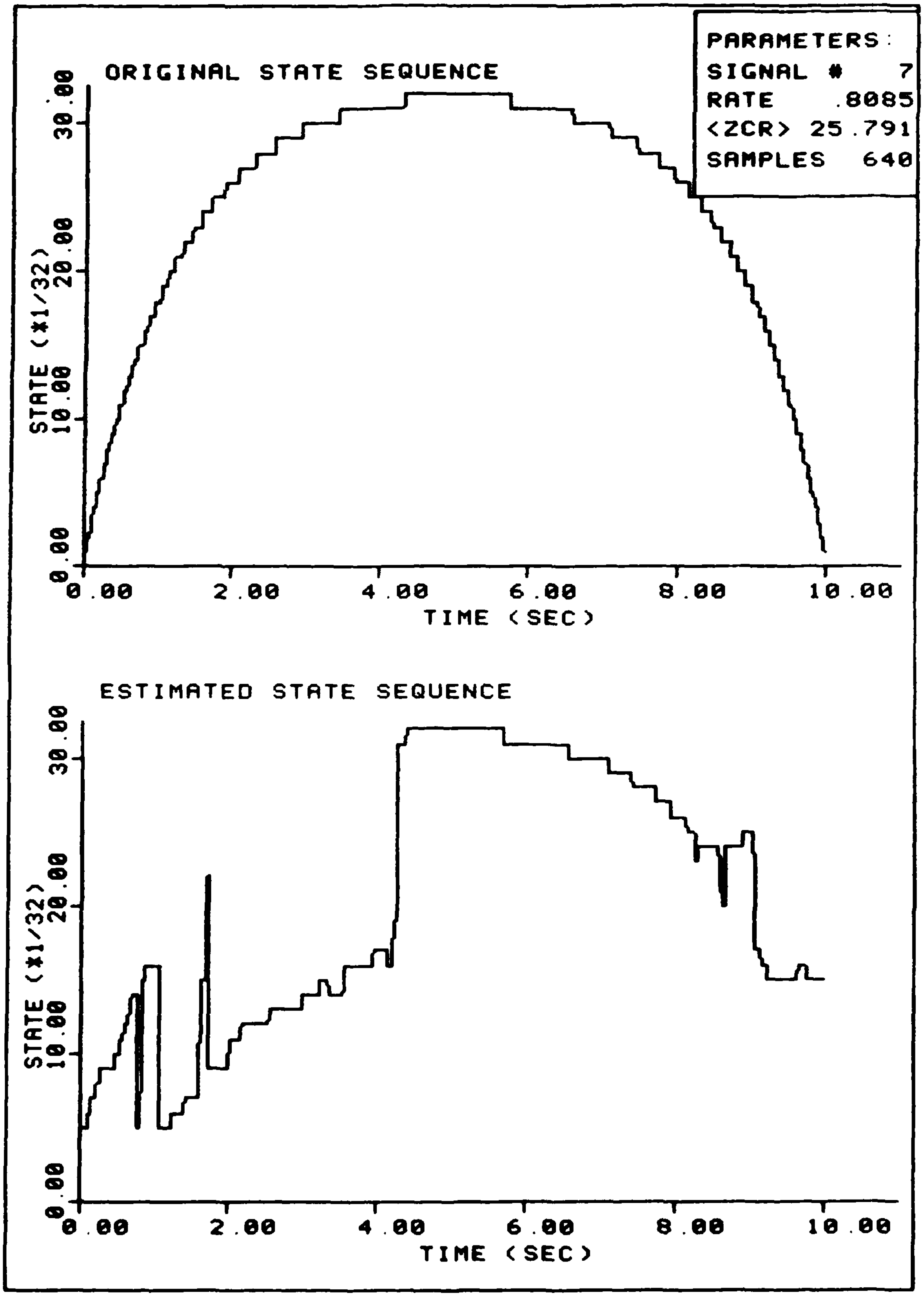


FIGURE 115. Sample Estimation

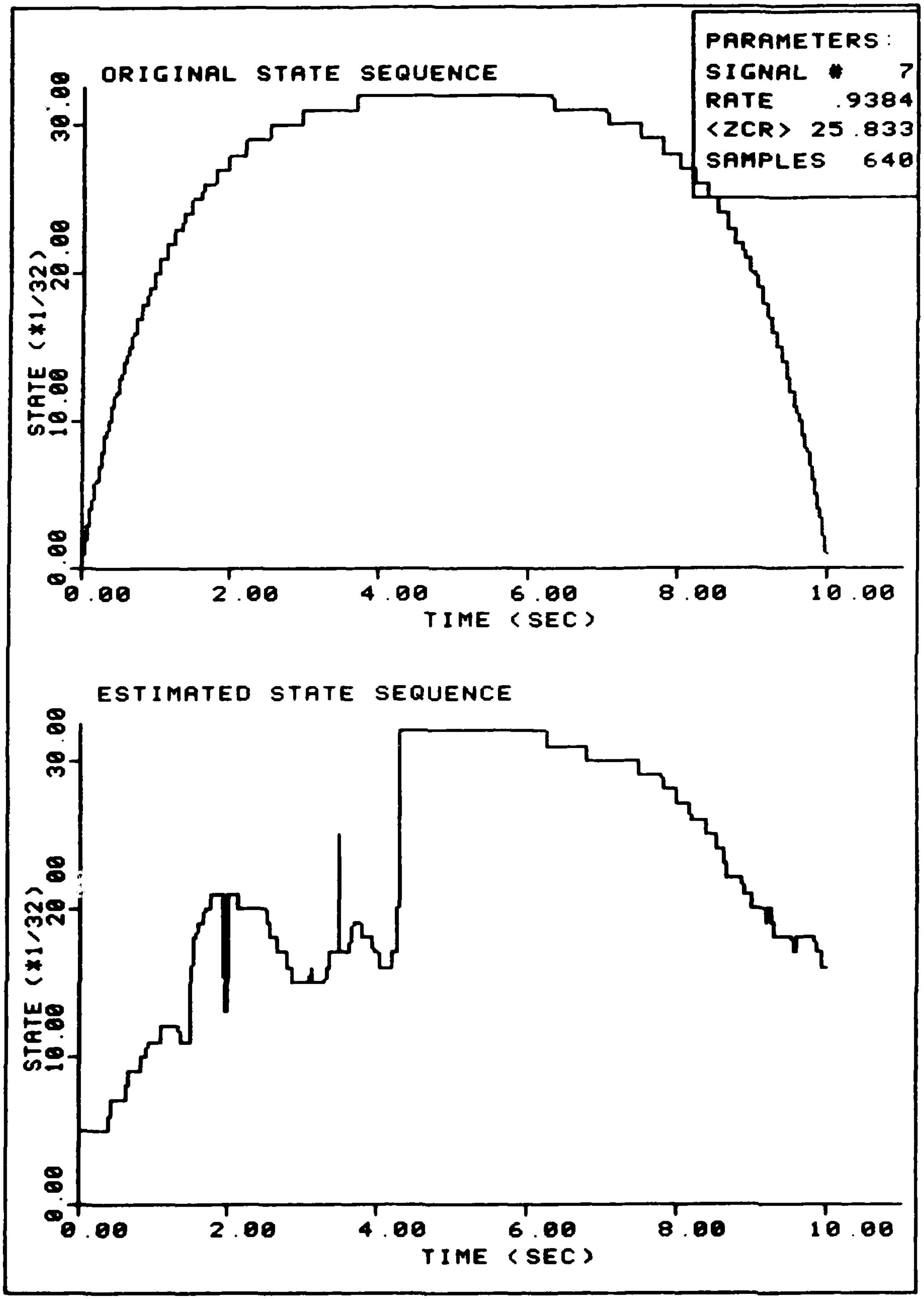


FIGURE 116. Sample Estimation

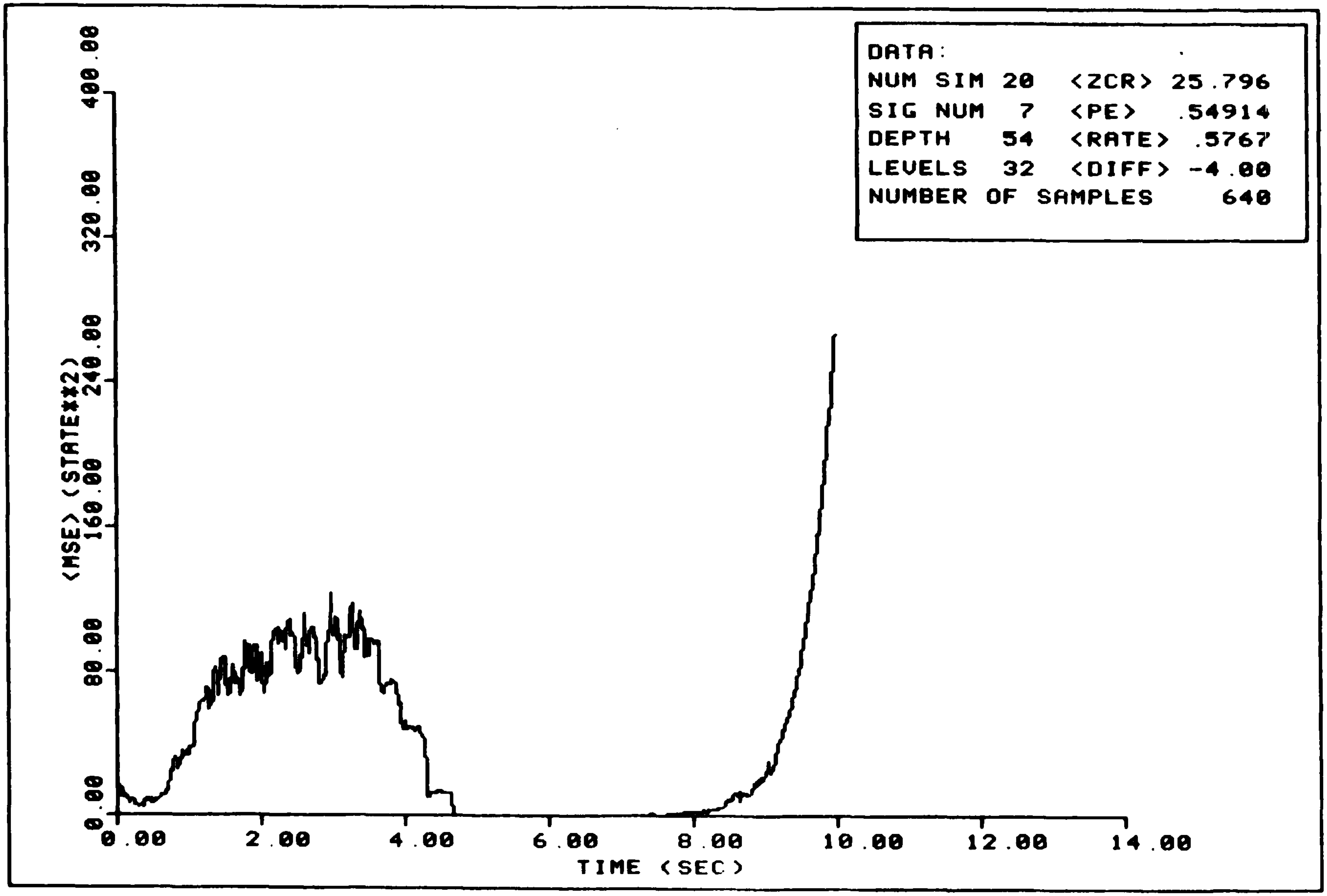


FIGURE 117. Ensemble Average MSE Performance

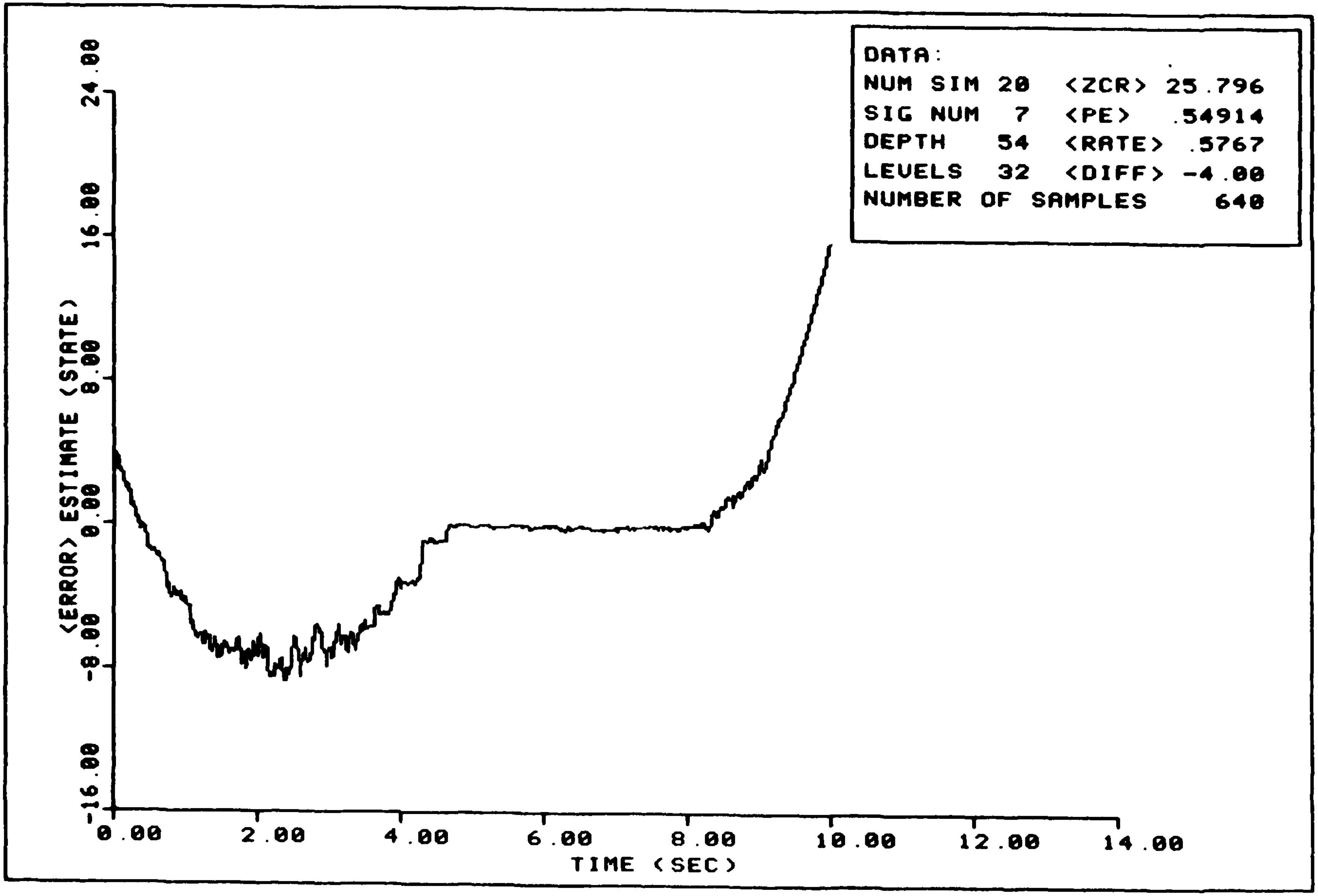


FIGURE 118. Ensemble Average State Error Performance

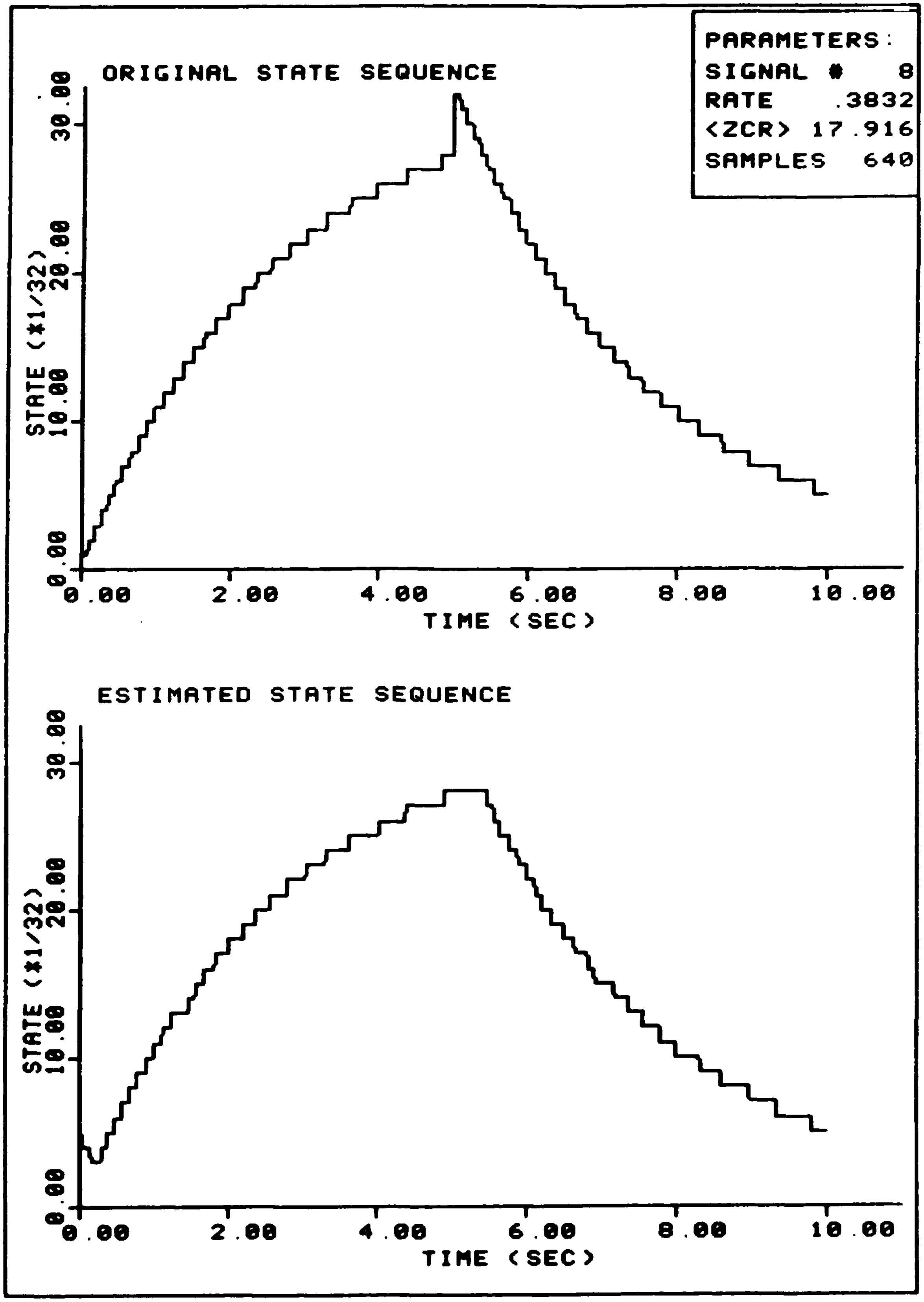


FIGURE 119. Sample Estimation

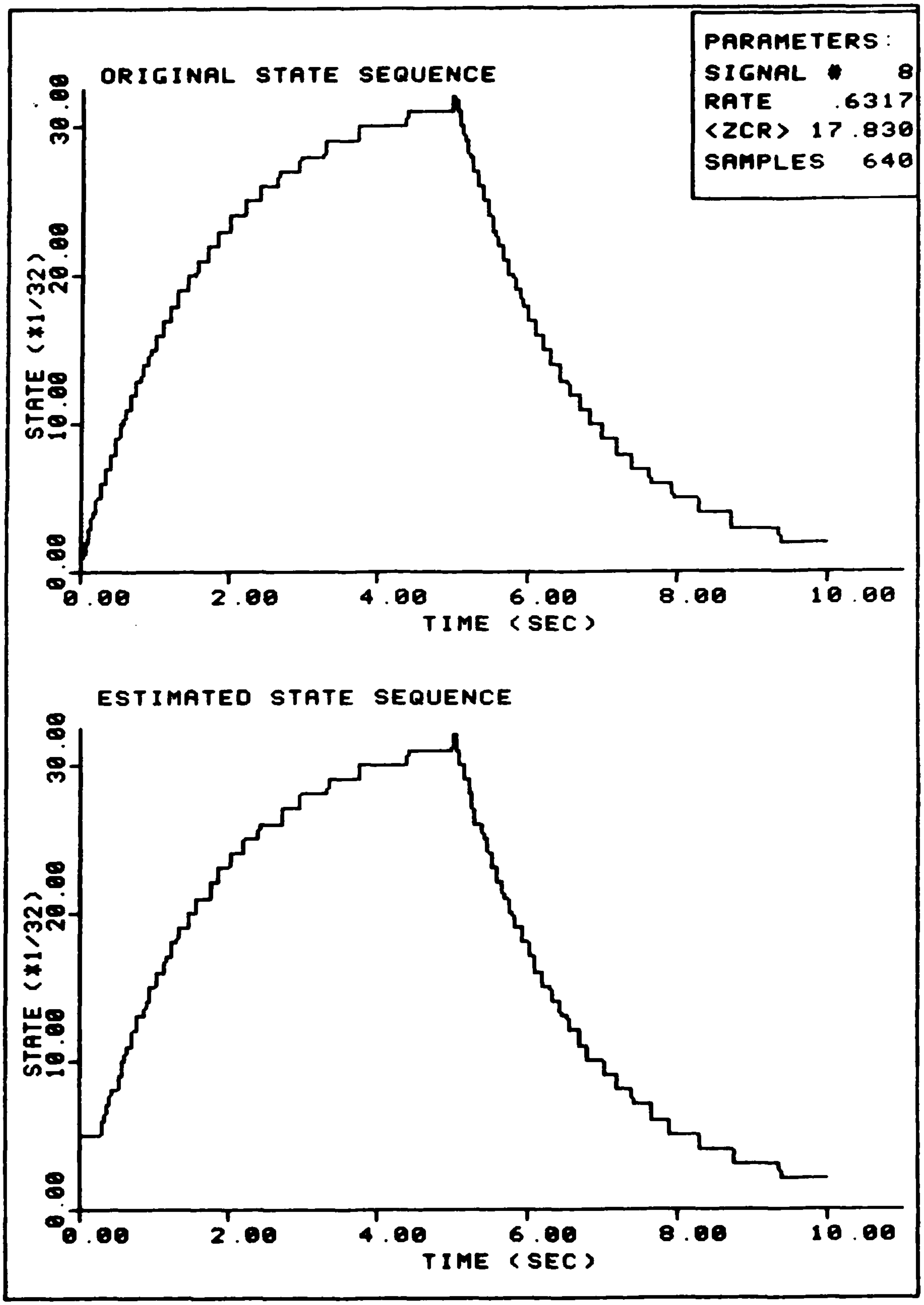


FIGURE 120. Sample Estimation

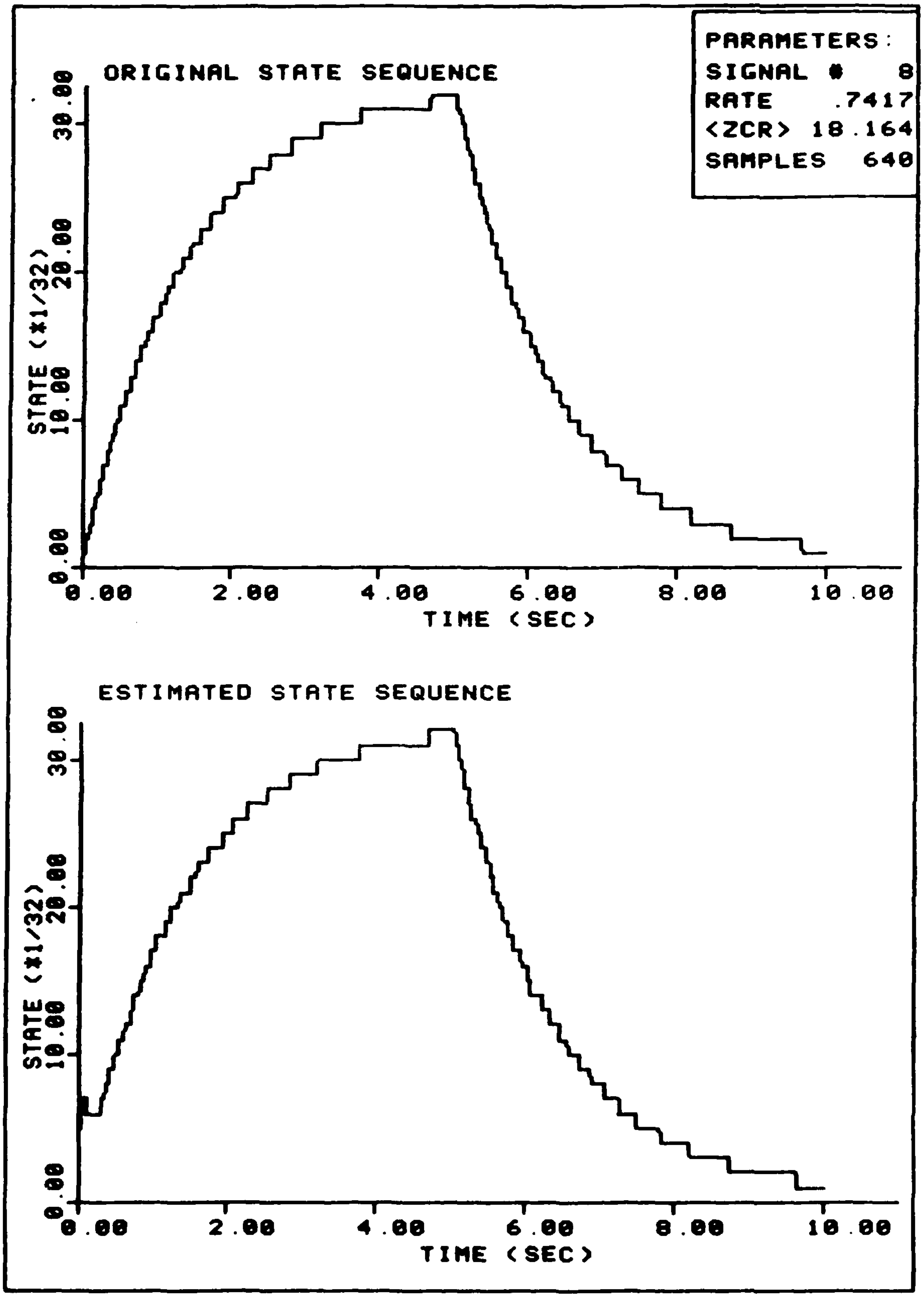


FIGURE 121. Sample Estimation

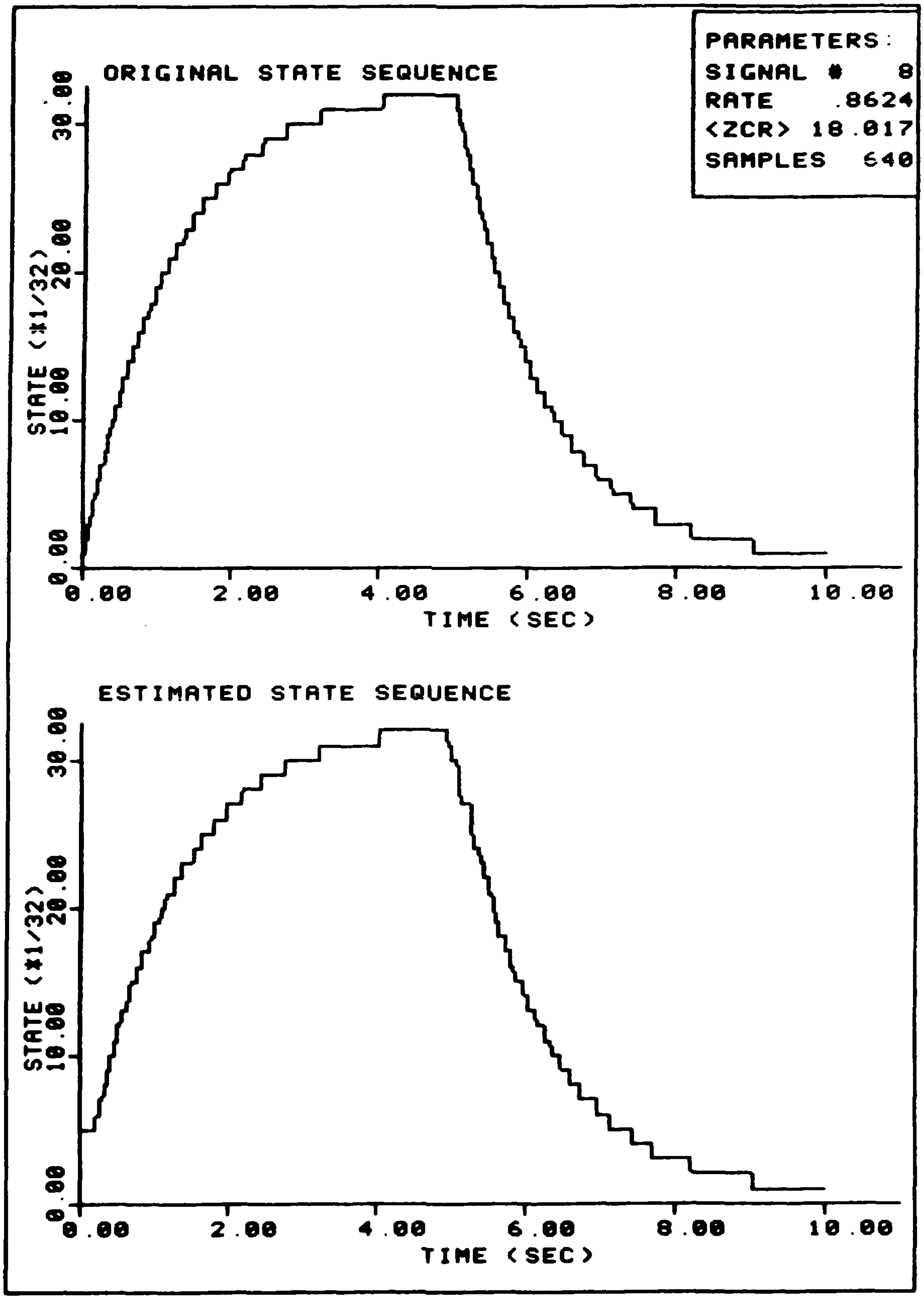


FIGURE 122. Sample Estimation

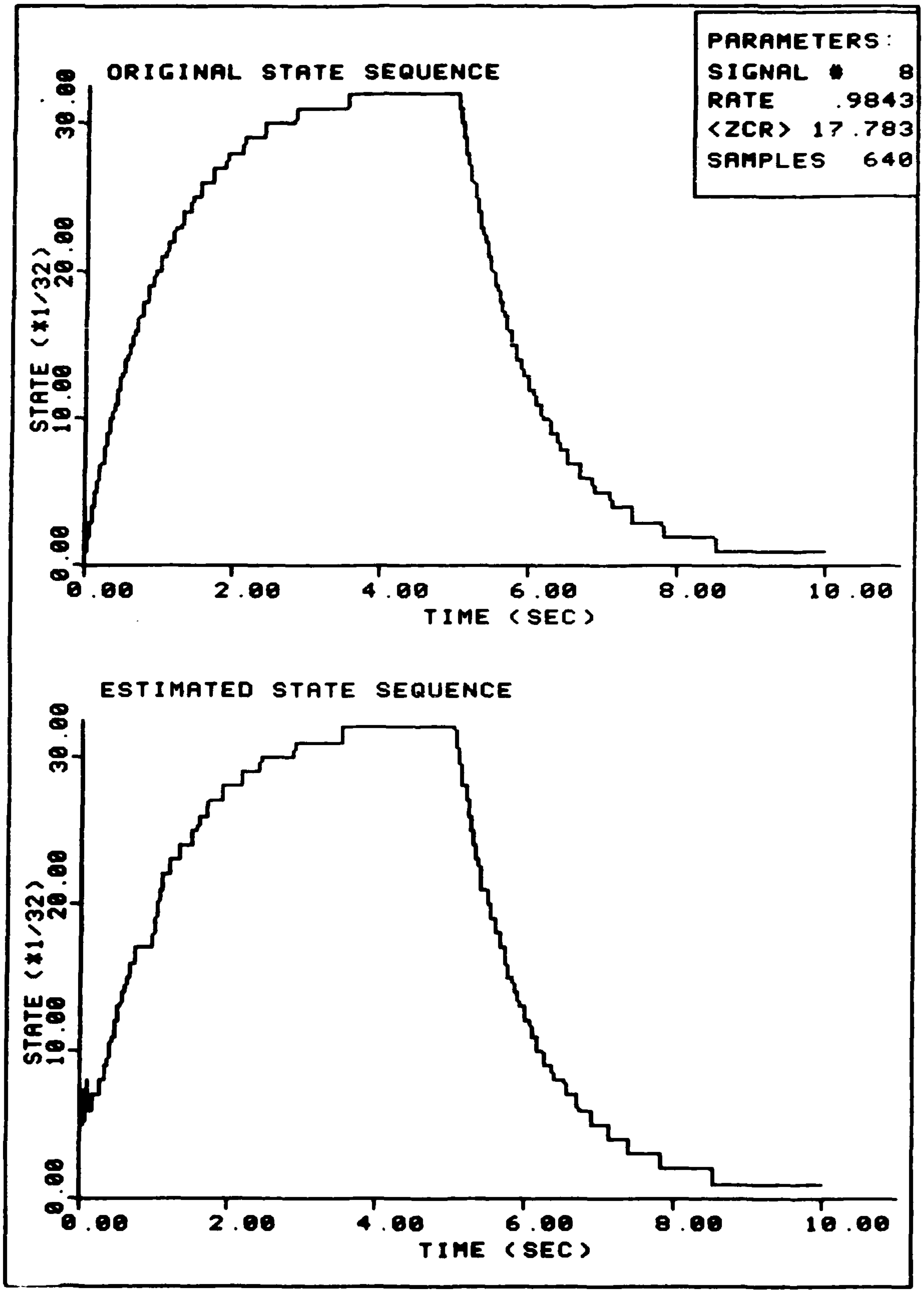


FIGURE 123. Sample Estimation

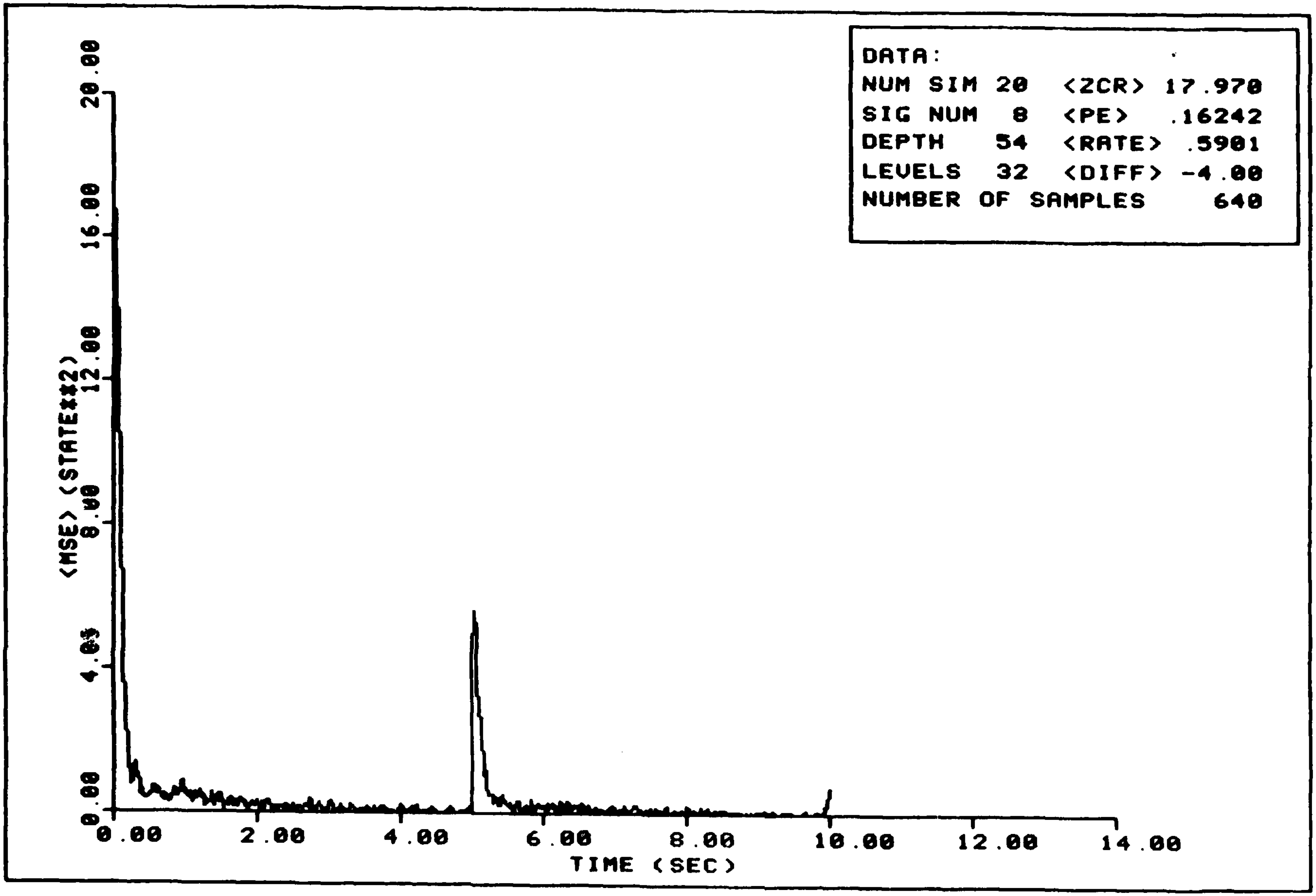


FIGURE 124. Ensemble Average MSE Performance

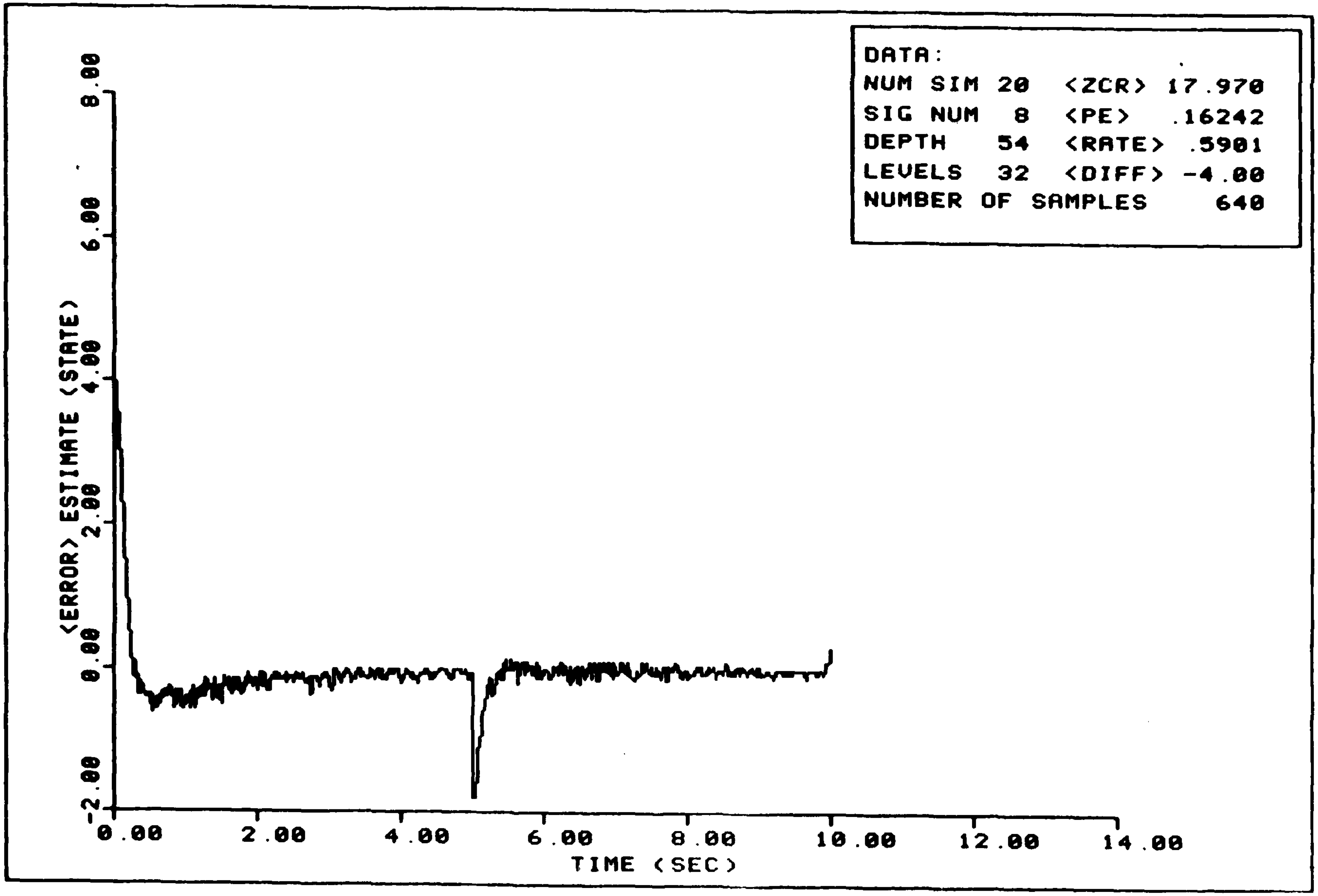


FIGURE 125. Ensemble Average State Error Performance

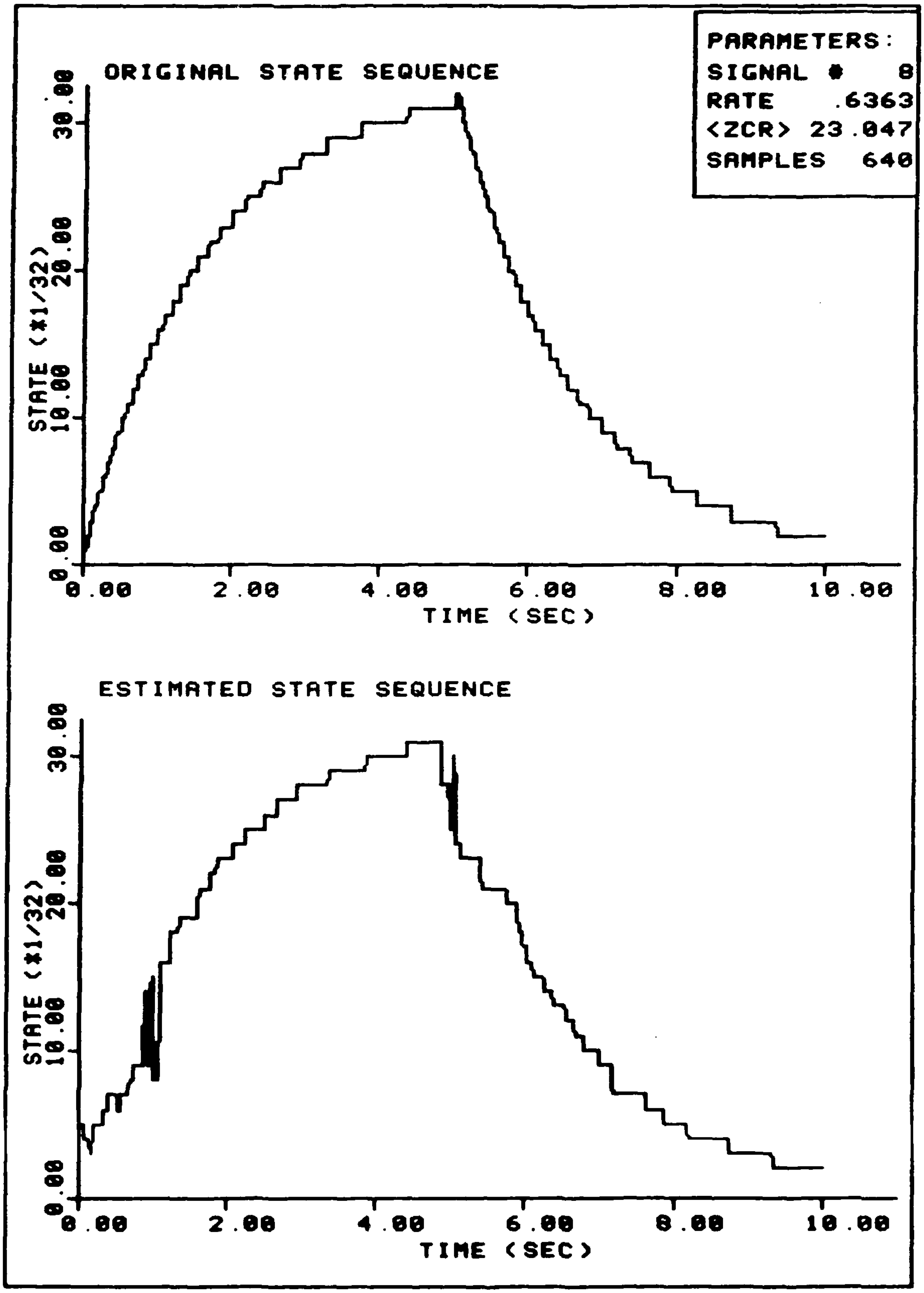


FIGURE 126. Sample Estimation

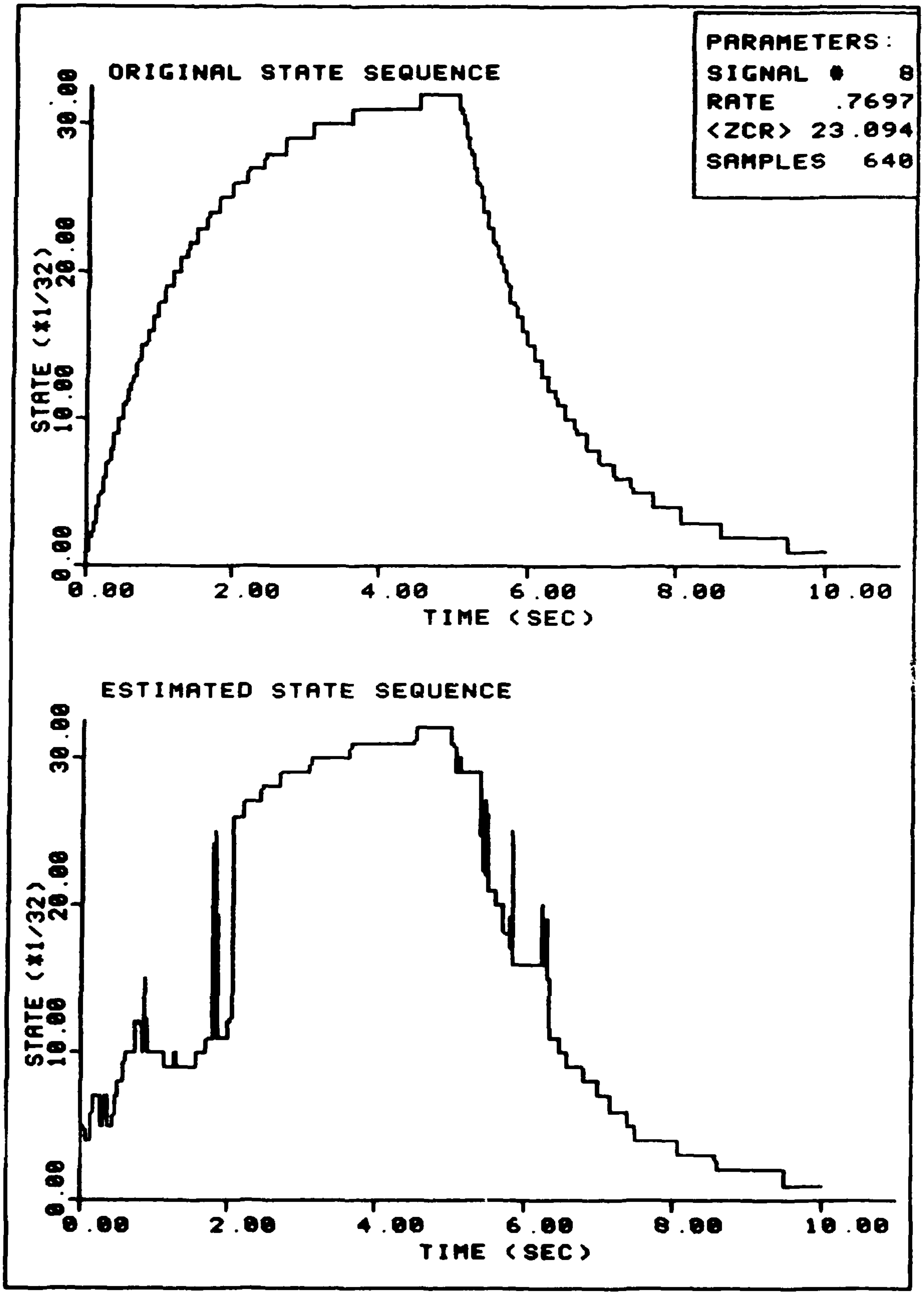


FIGURE 127. Sample Estimation

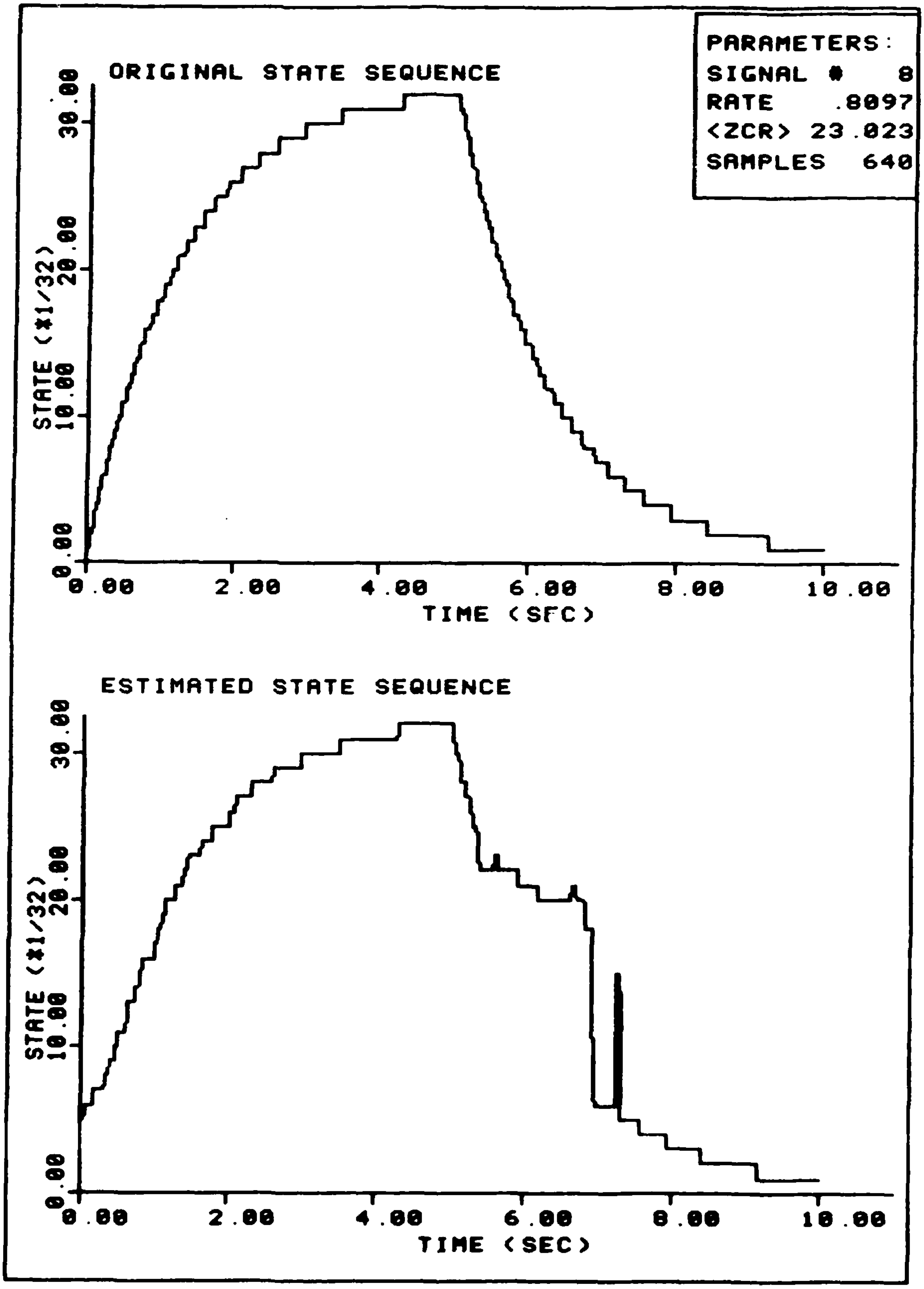


FIGURE 128. Sample Estimation

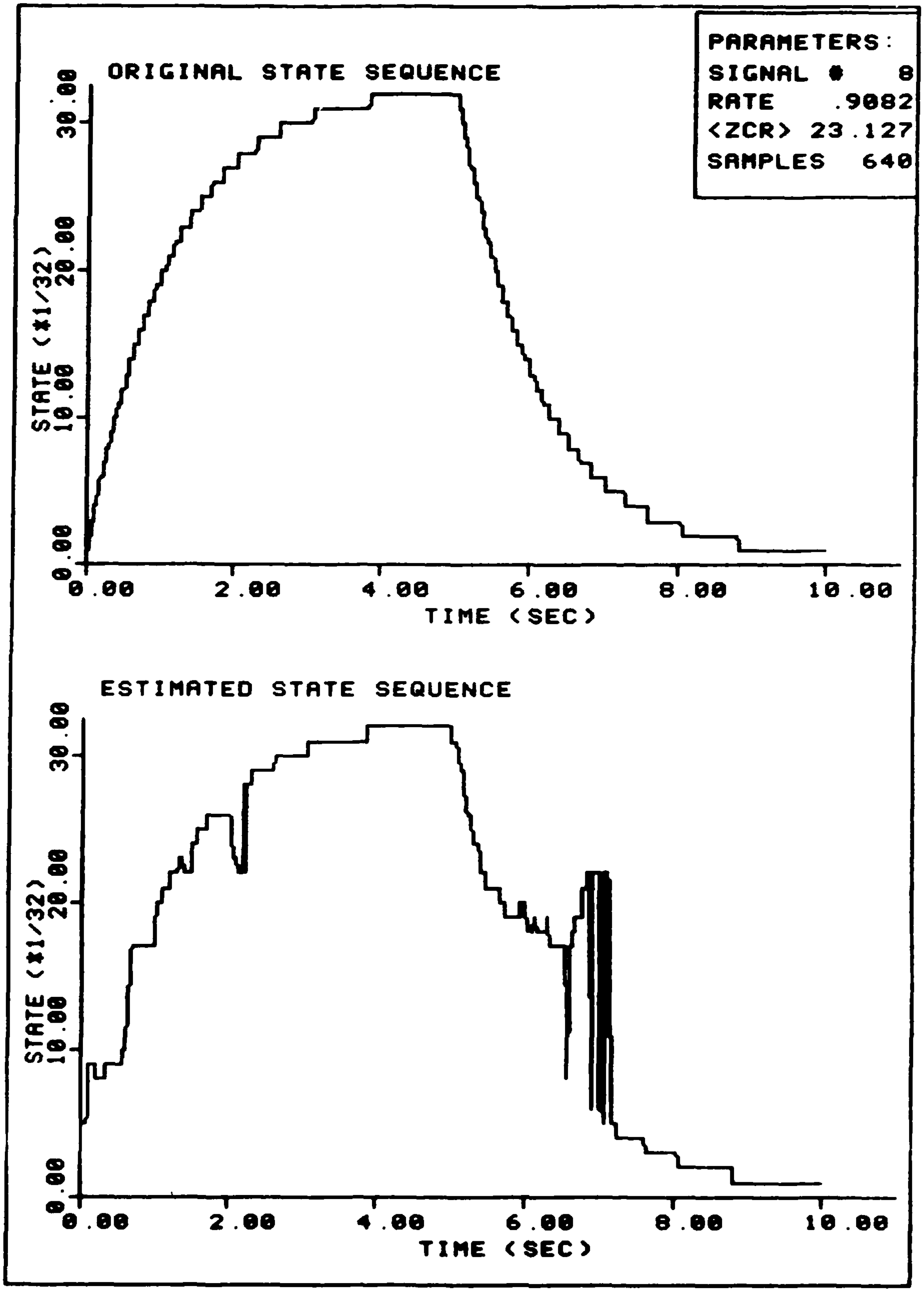


FIGURE 129. Sample Estimation

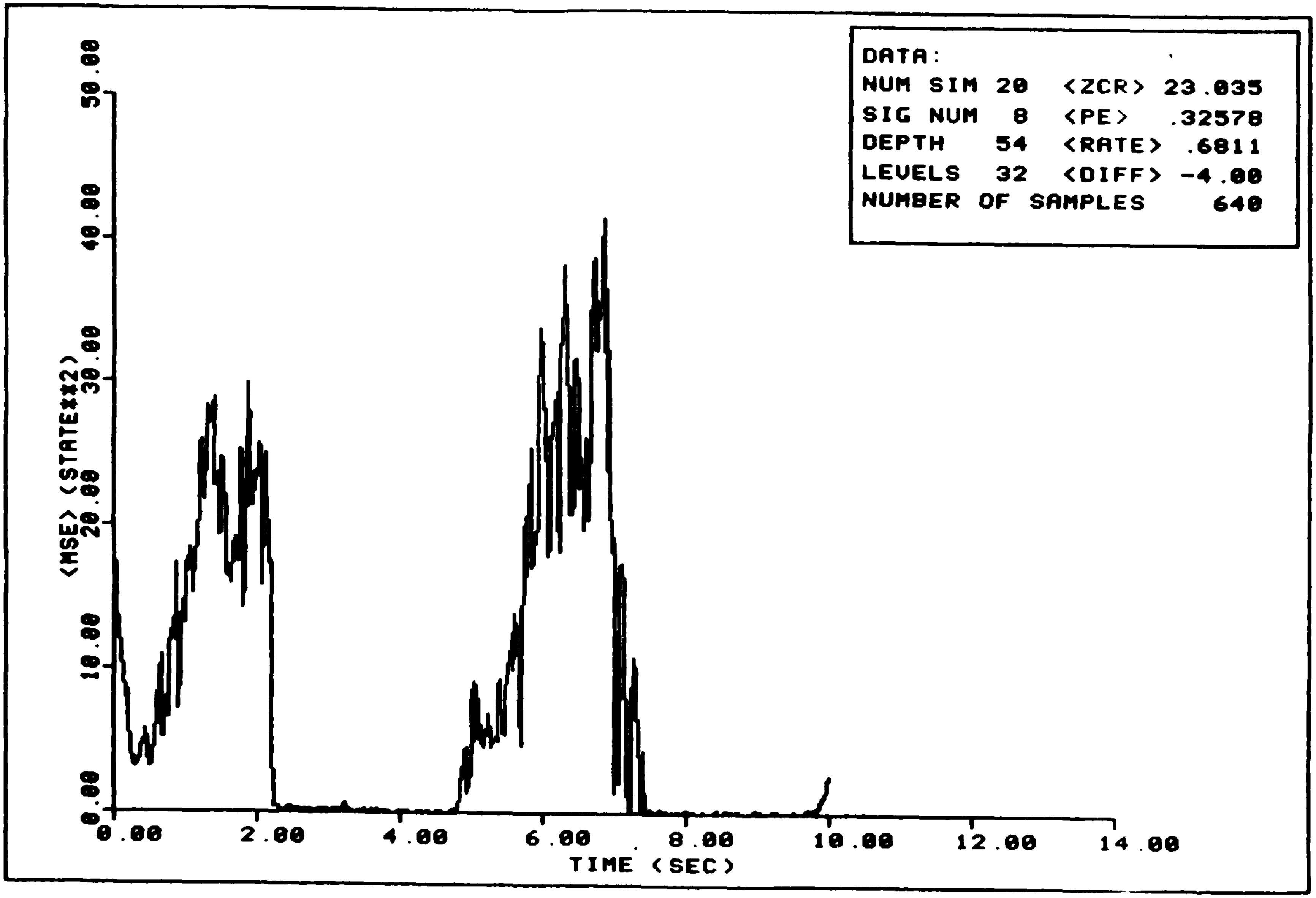


FIGURE 130. Ensemble Average MSE Performance

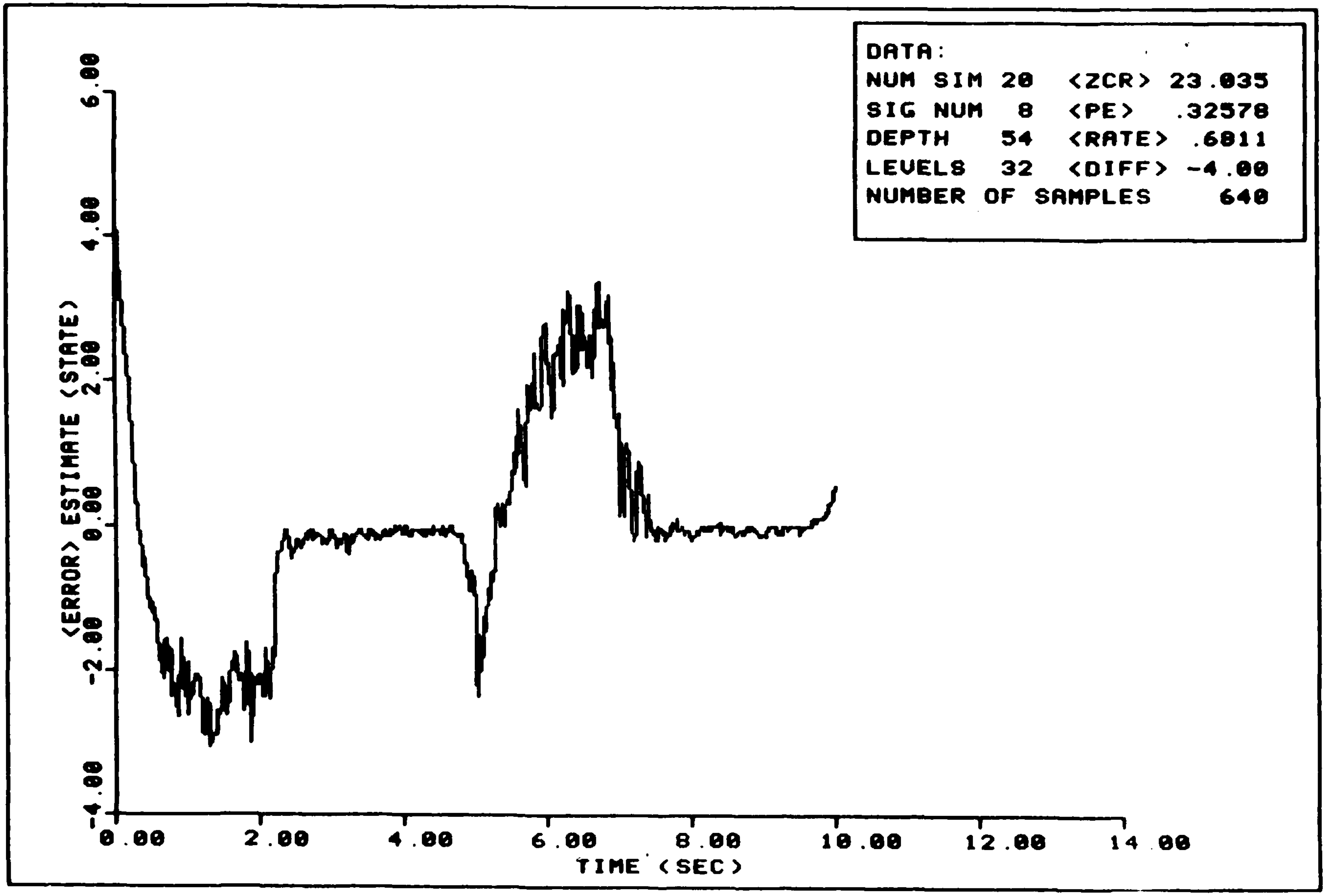


FIGURE 131. Ensemble Average State Error Performance

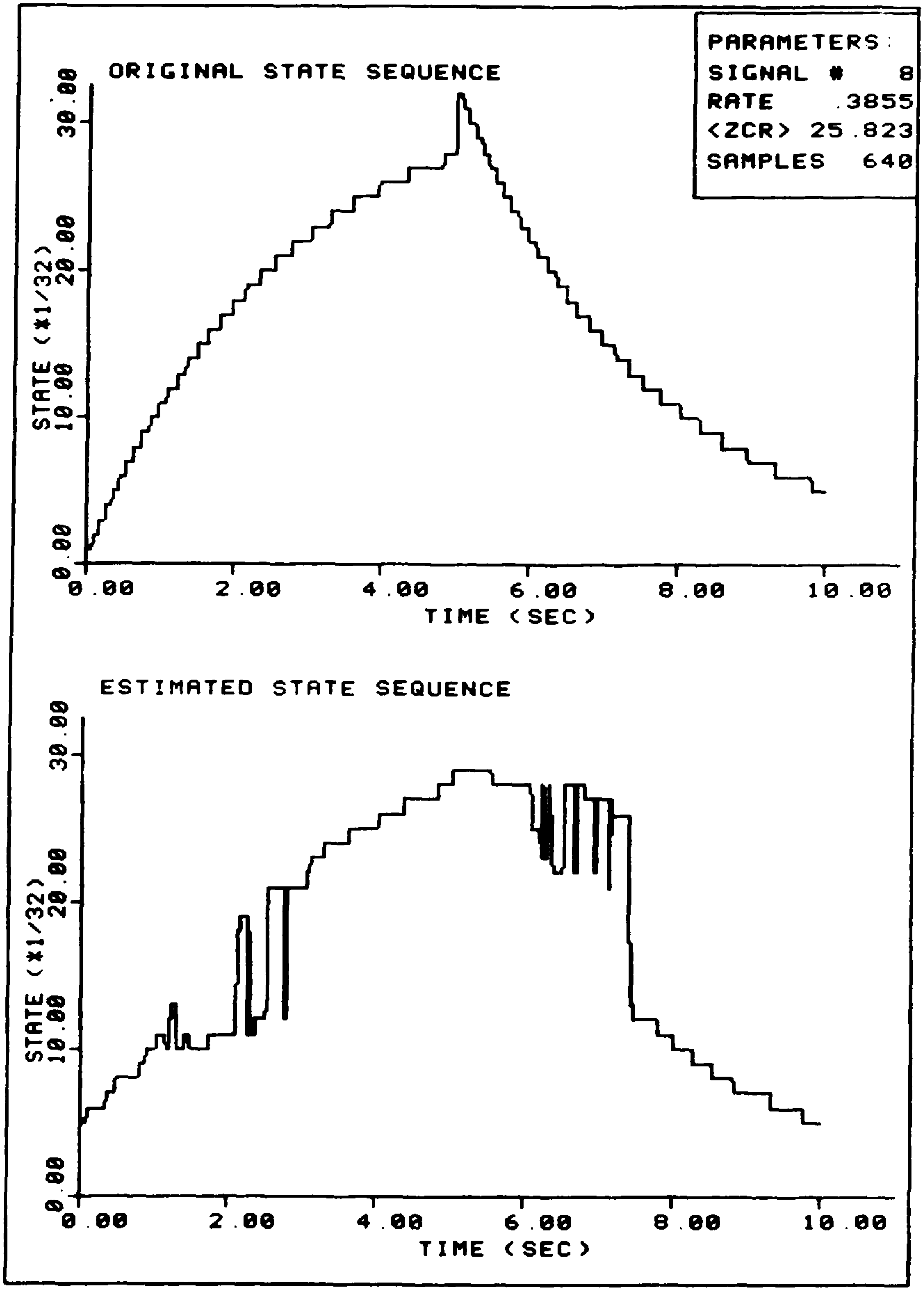


FIGURE 132. Sample Estimation

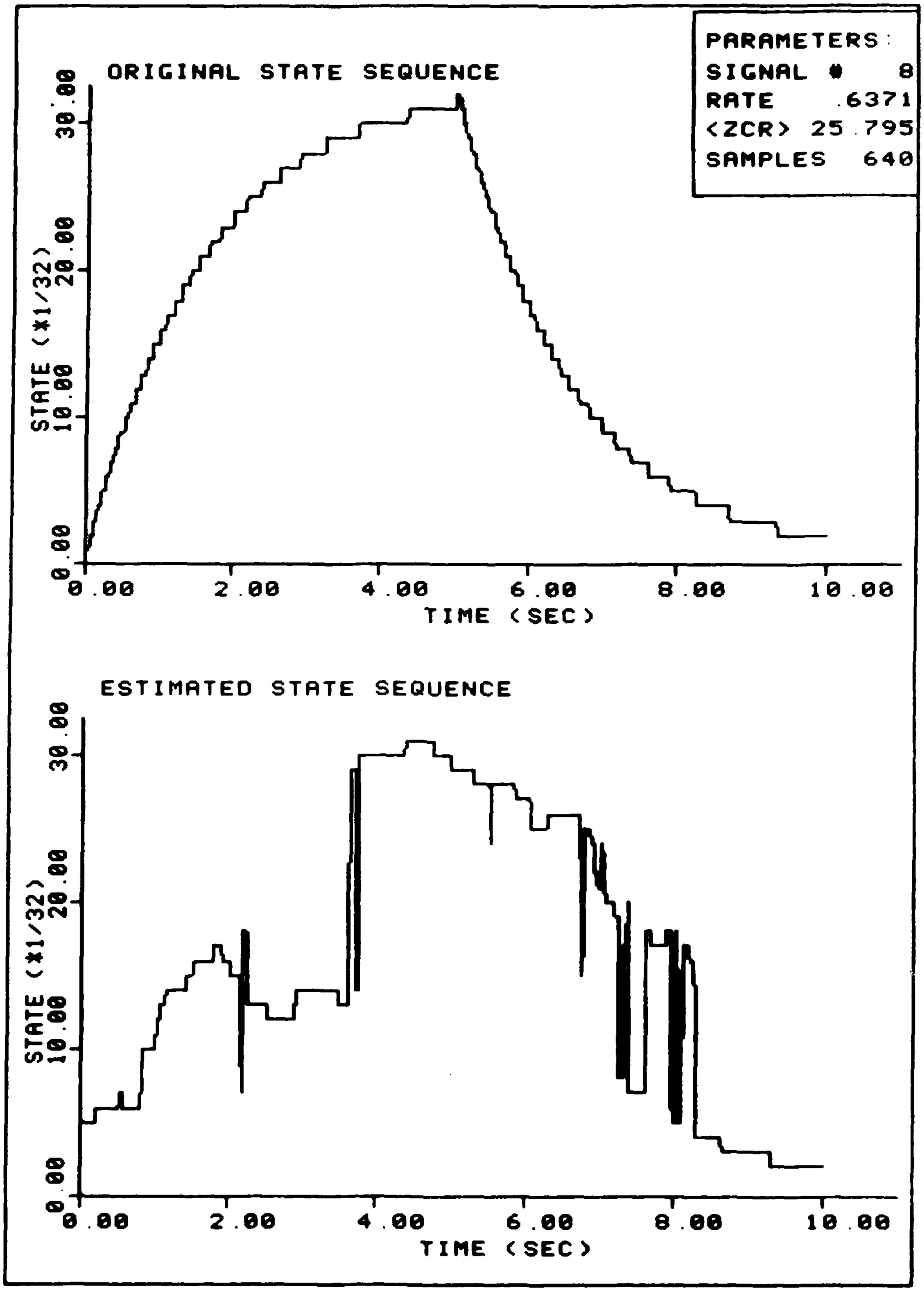


FIGURE 133. Sample Estimation

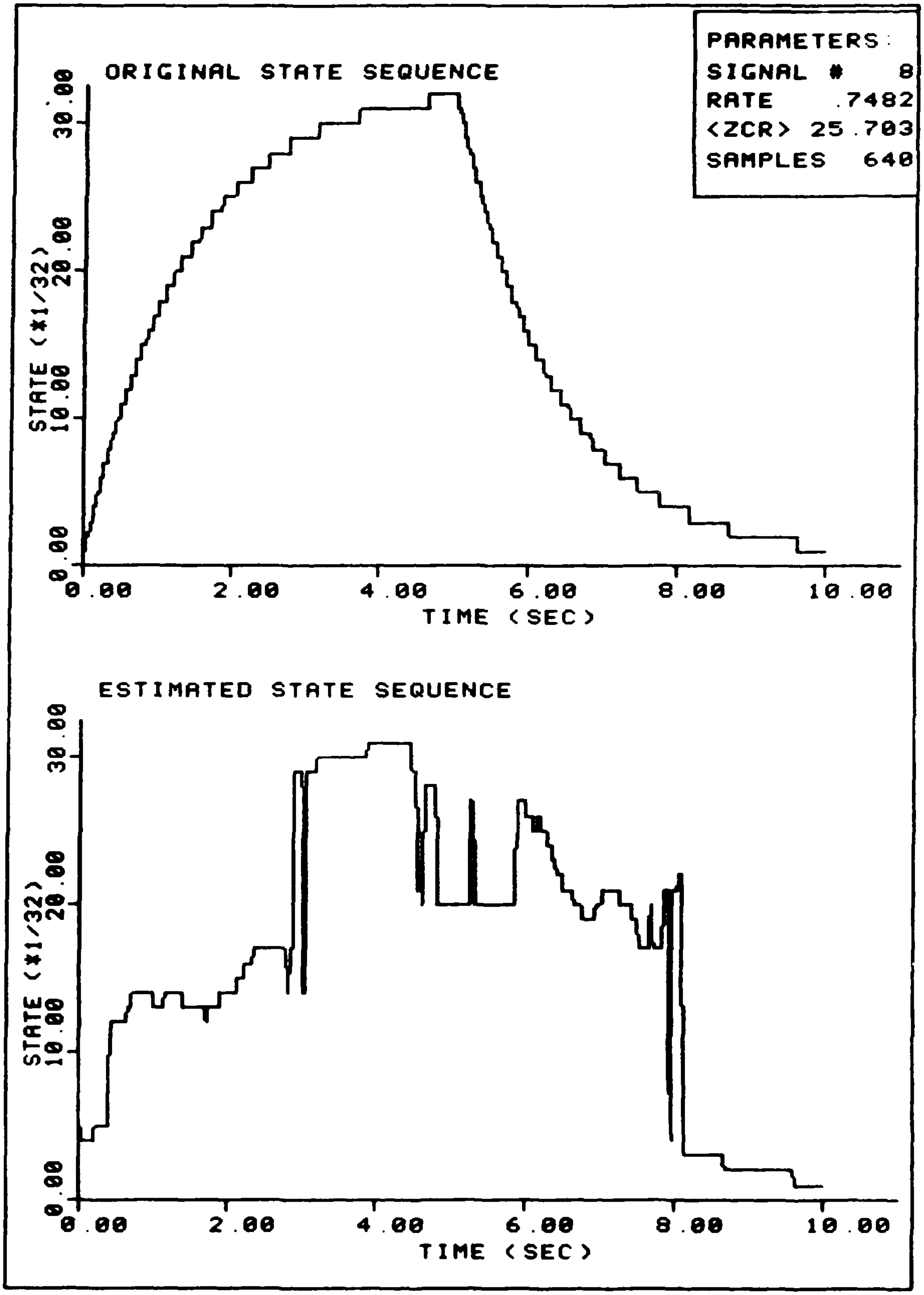


FIGURE 134. Sample Estimation

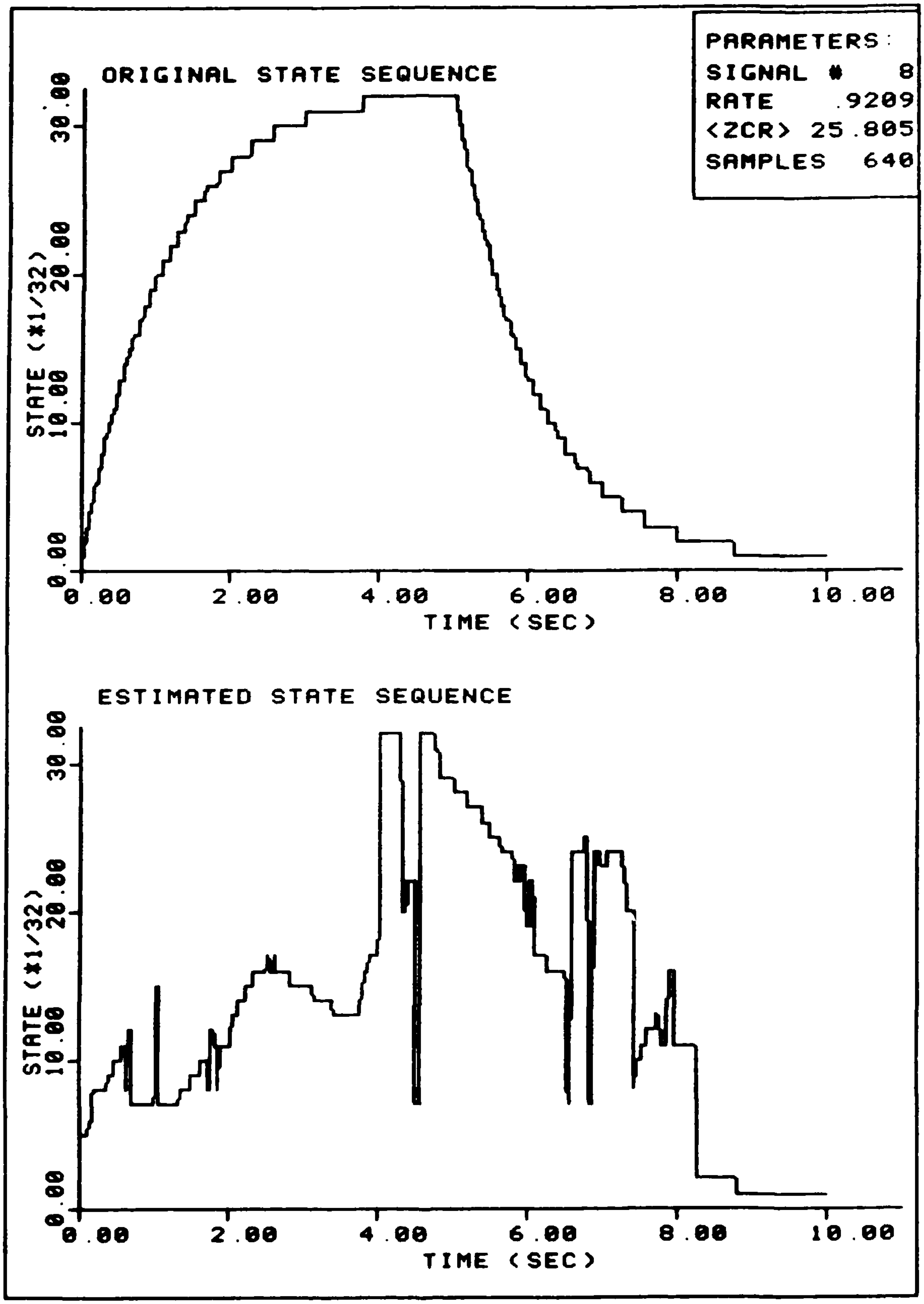


FIGURE 135. Sample Estimation

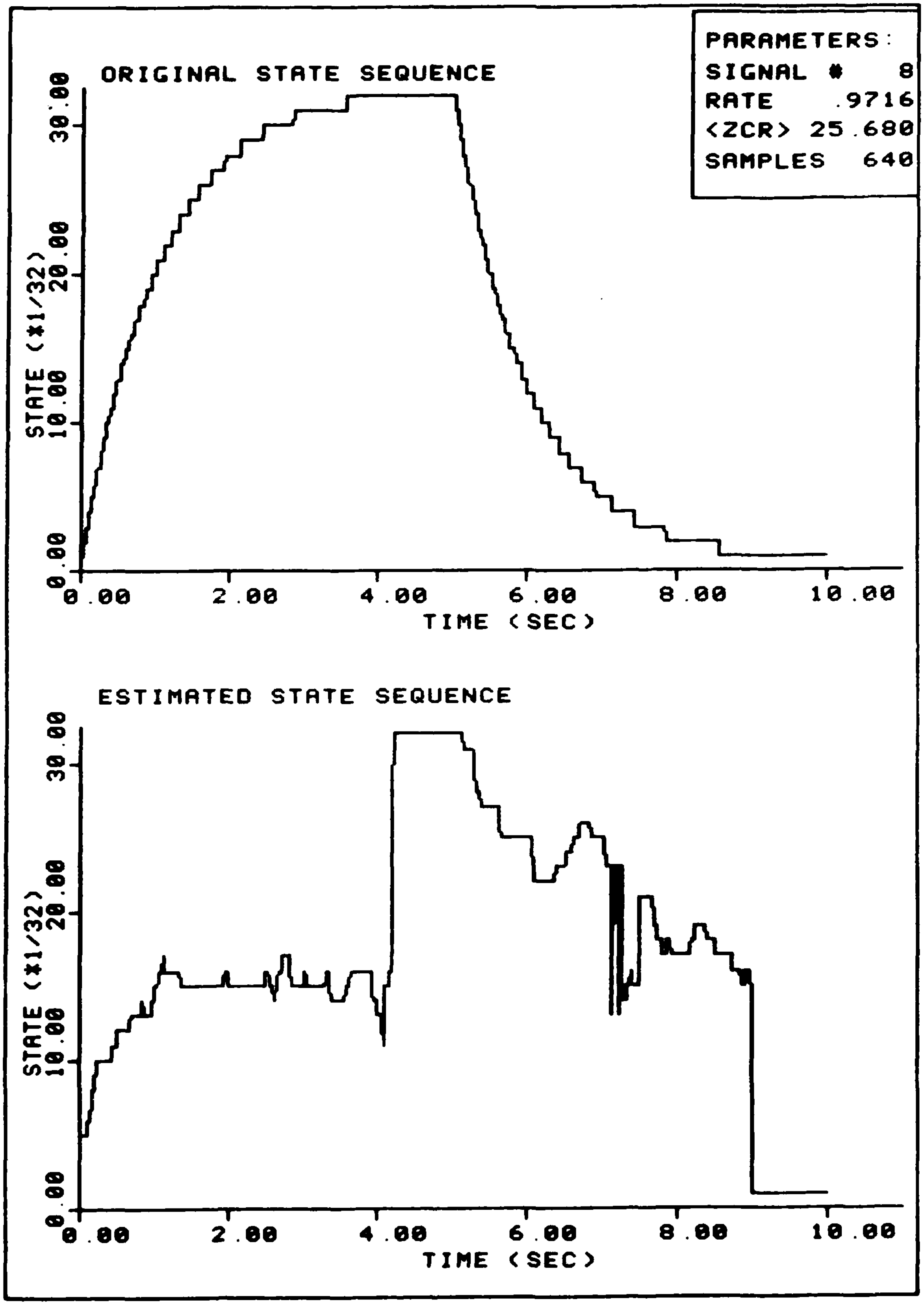


FIGURE 136. Sample Estimation

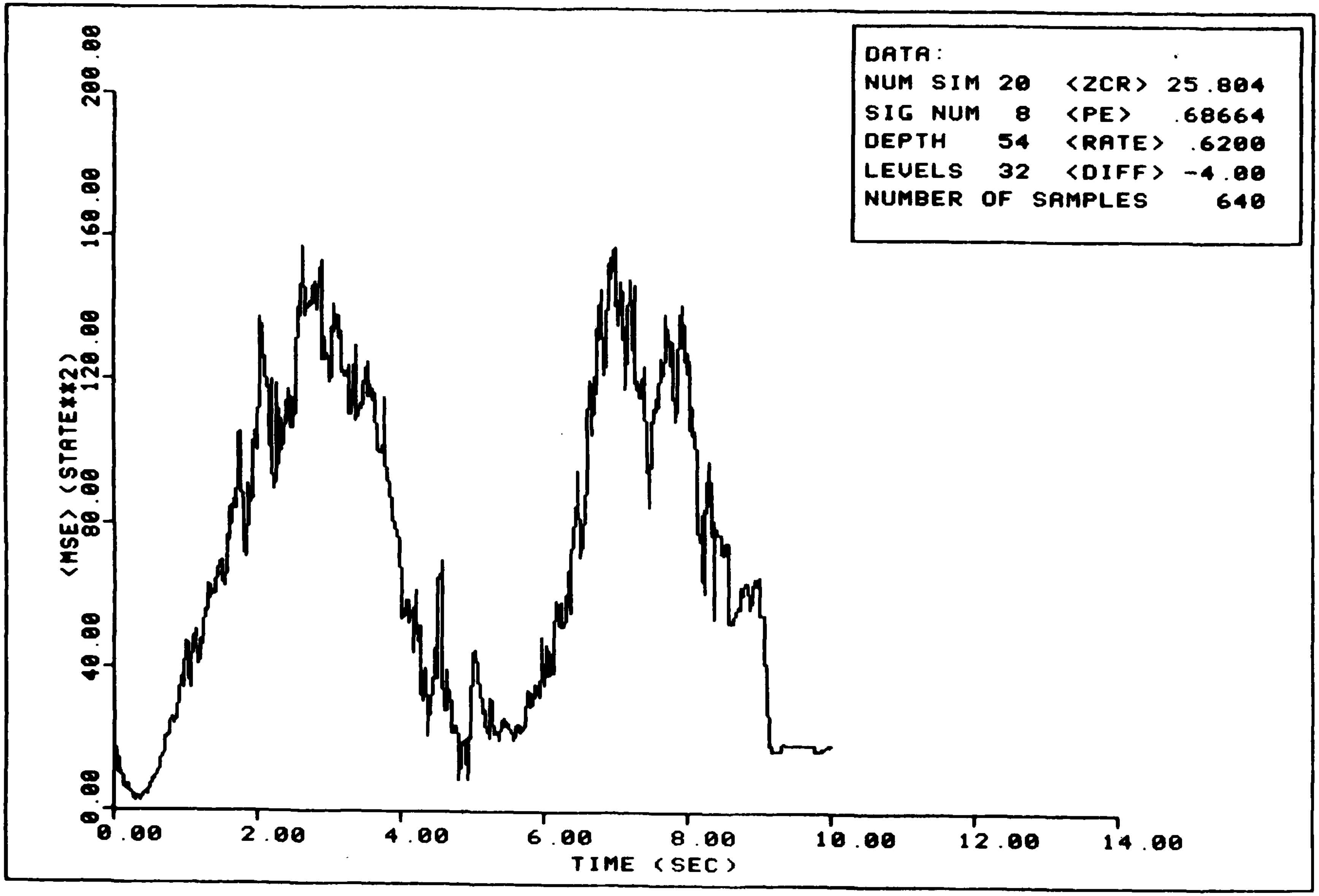


FIGURE 137. Ensemble Average MSE Performance

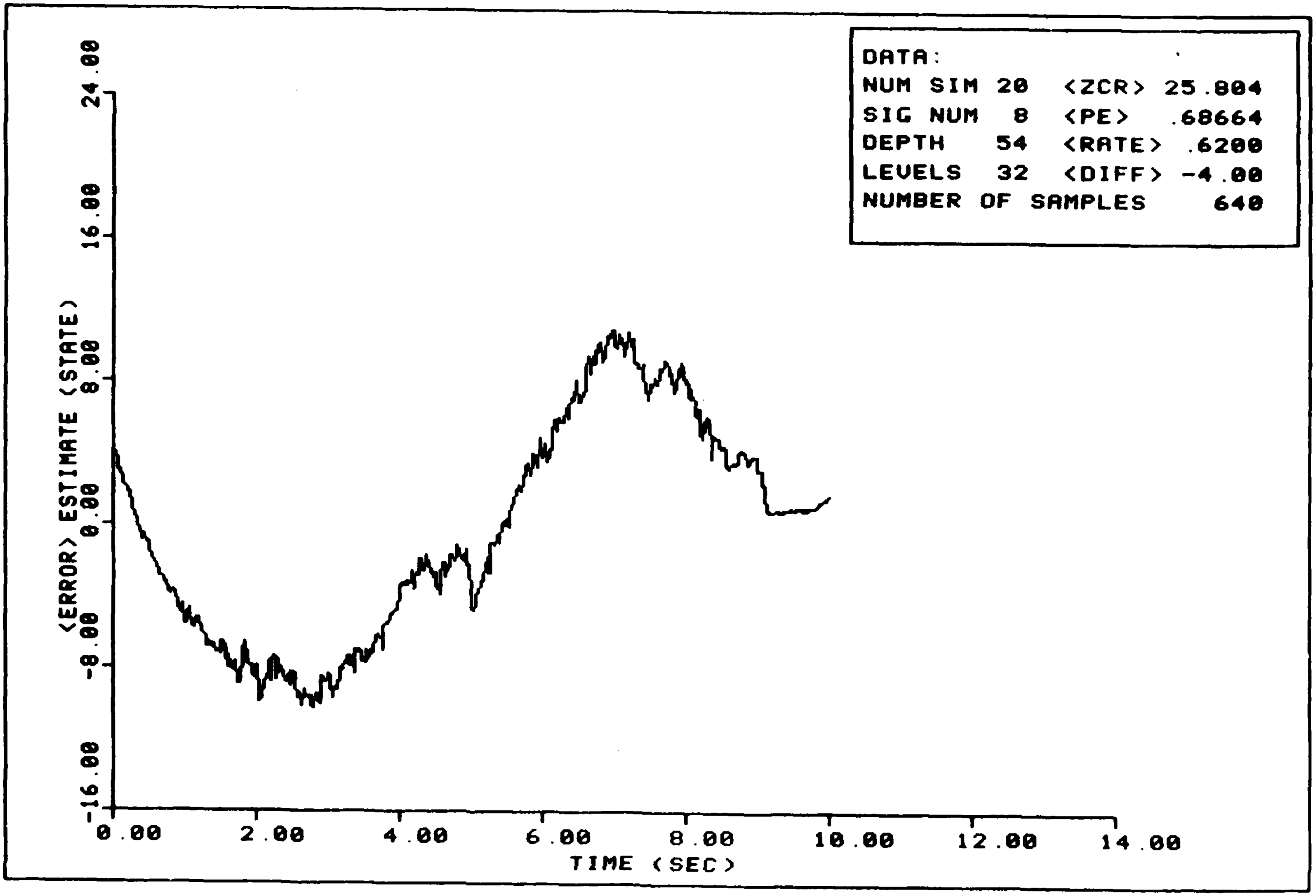


FIGURE 138. Ensemble Average State Error Performance

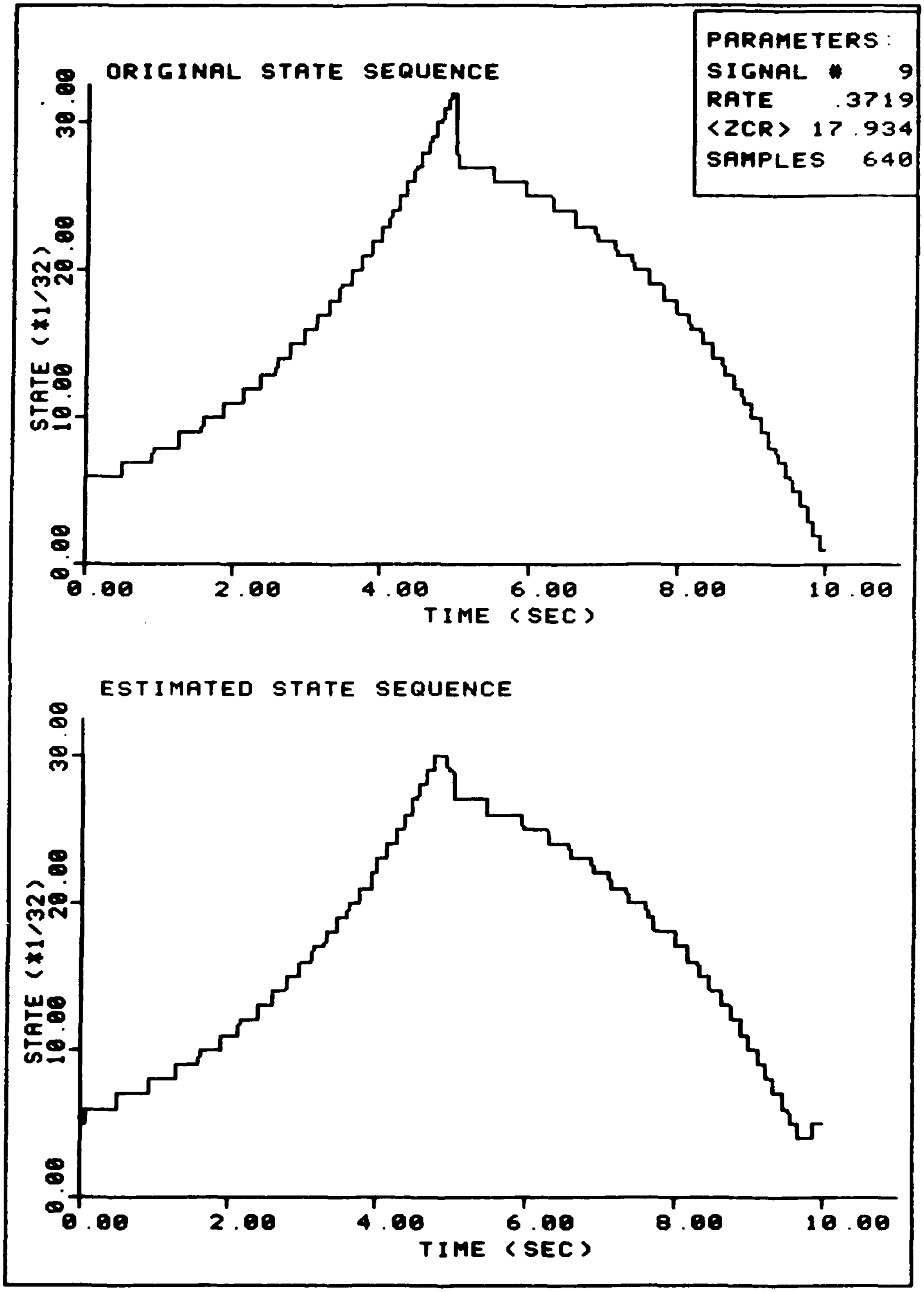


FIGURE 139. Sample Estimation

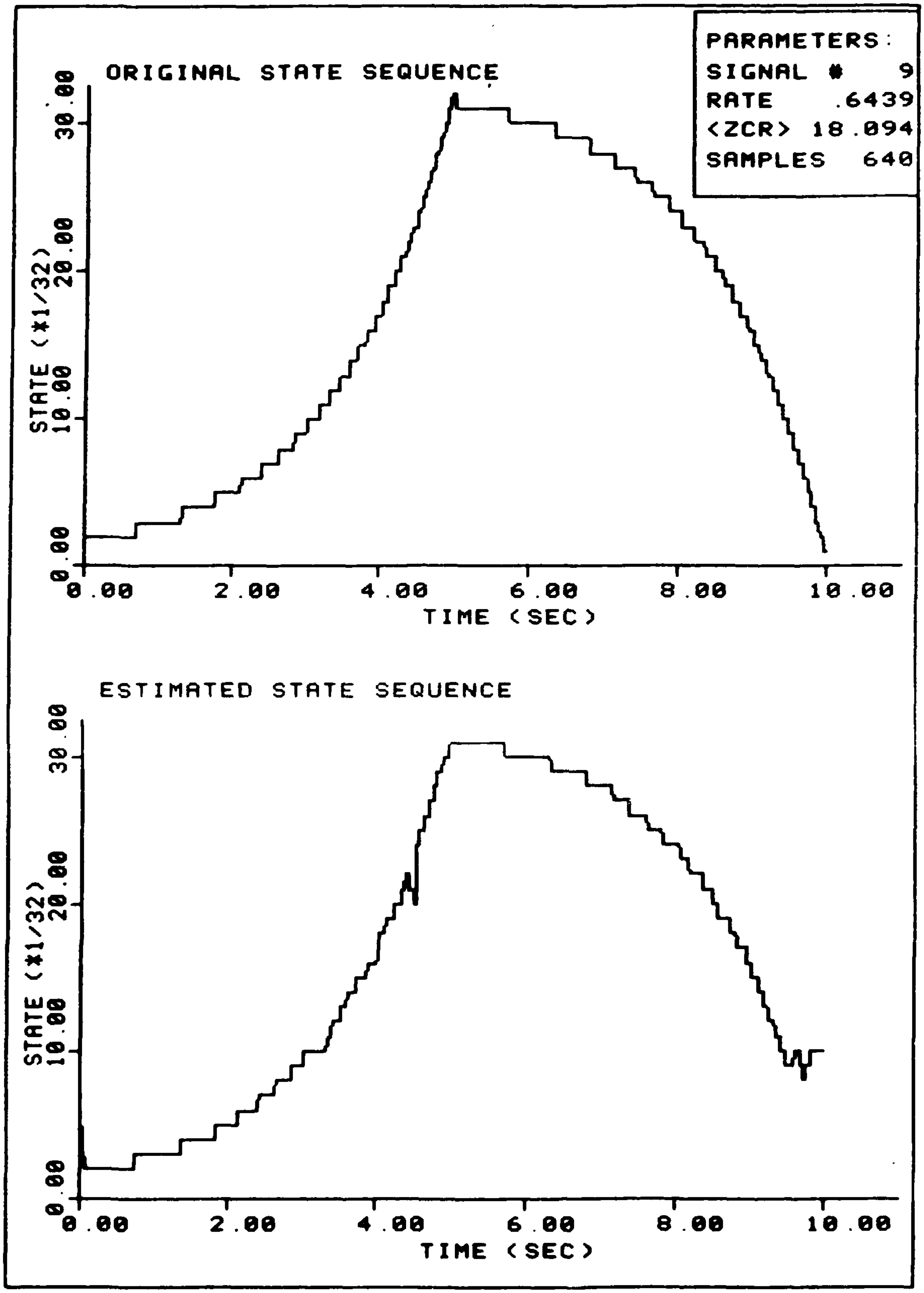


FIGURE 140. Sample Estimation

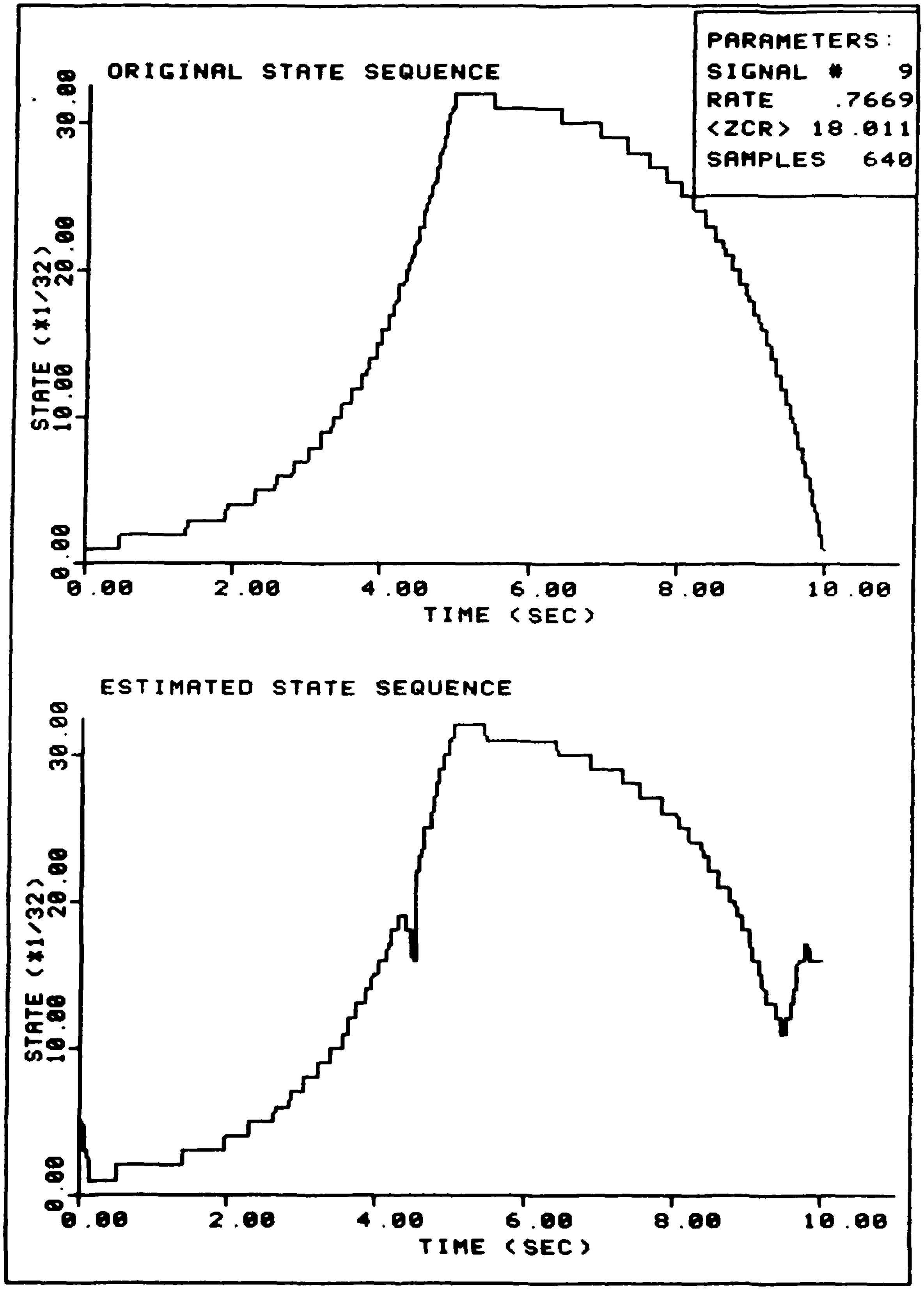


FIGURE 141. Sample Estimation

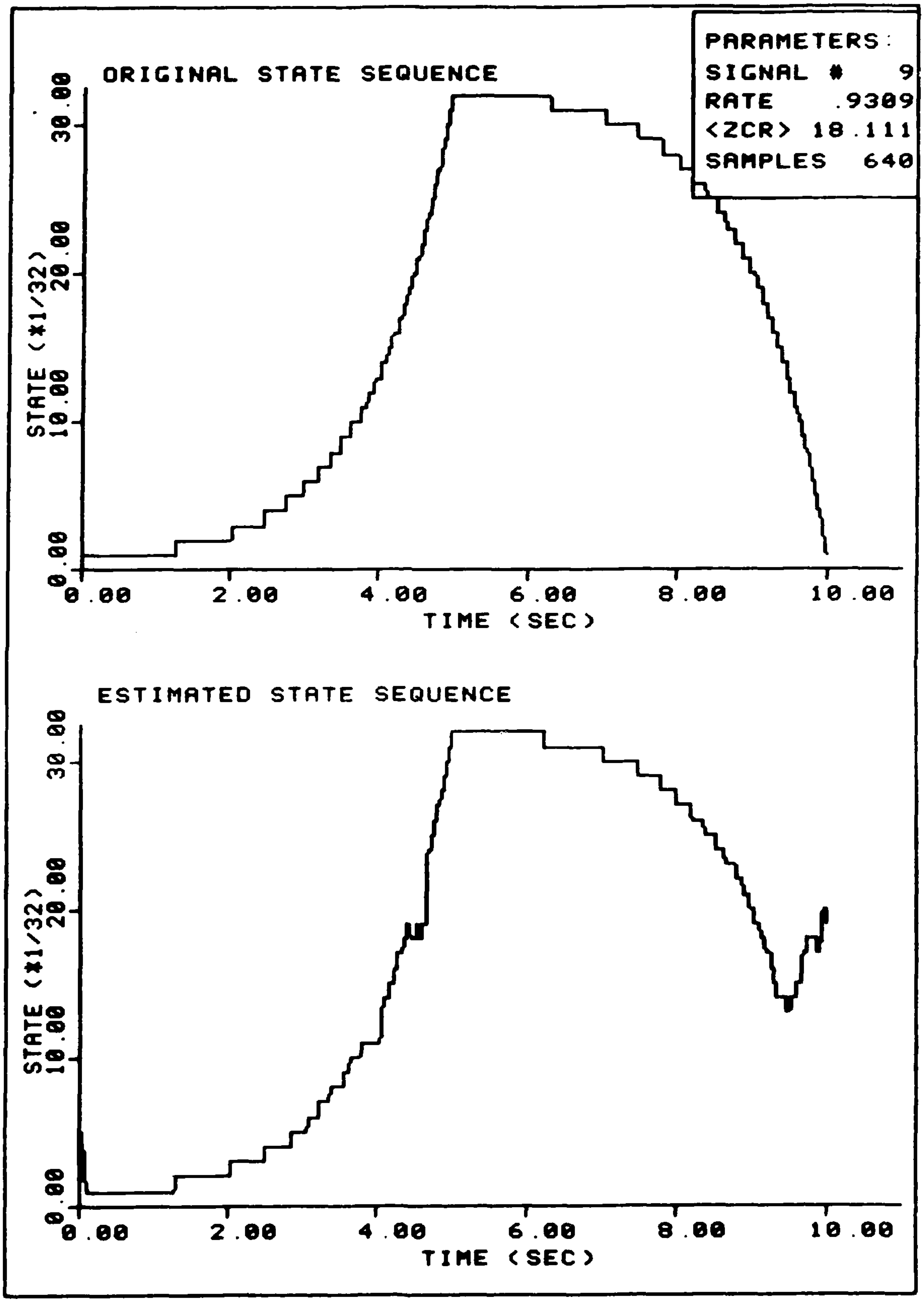


FIGURE 142. Sample Estimation

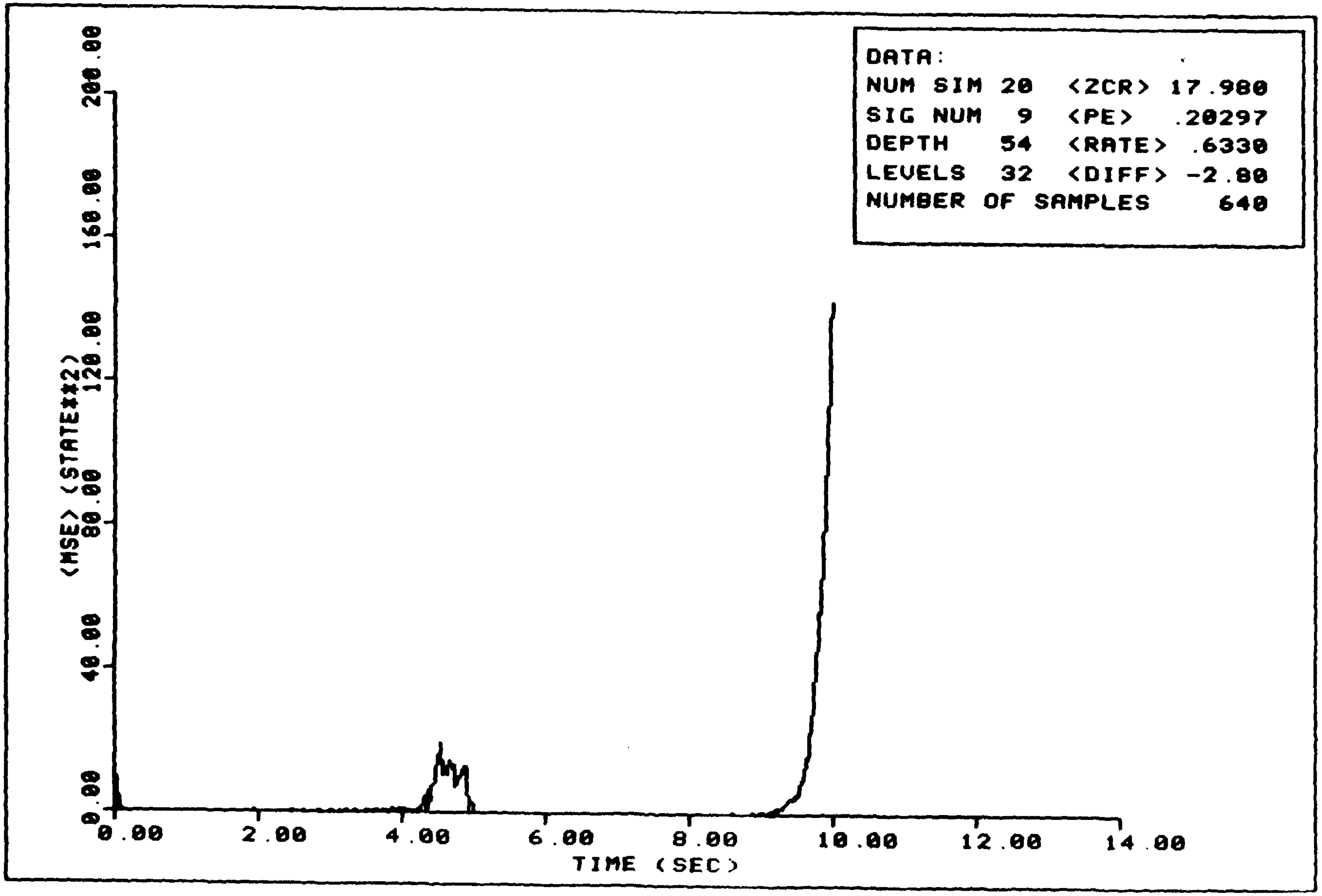


FIGURE 143. Ensemble Average MSE Performance

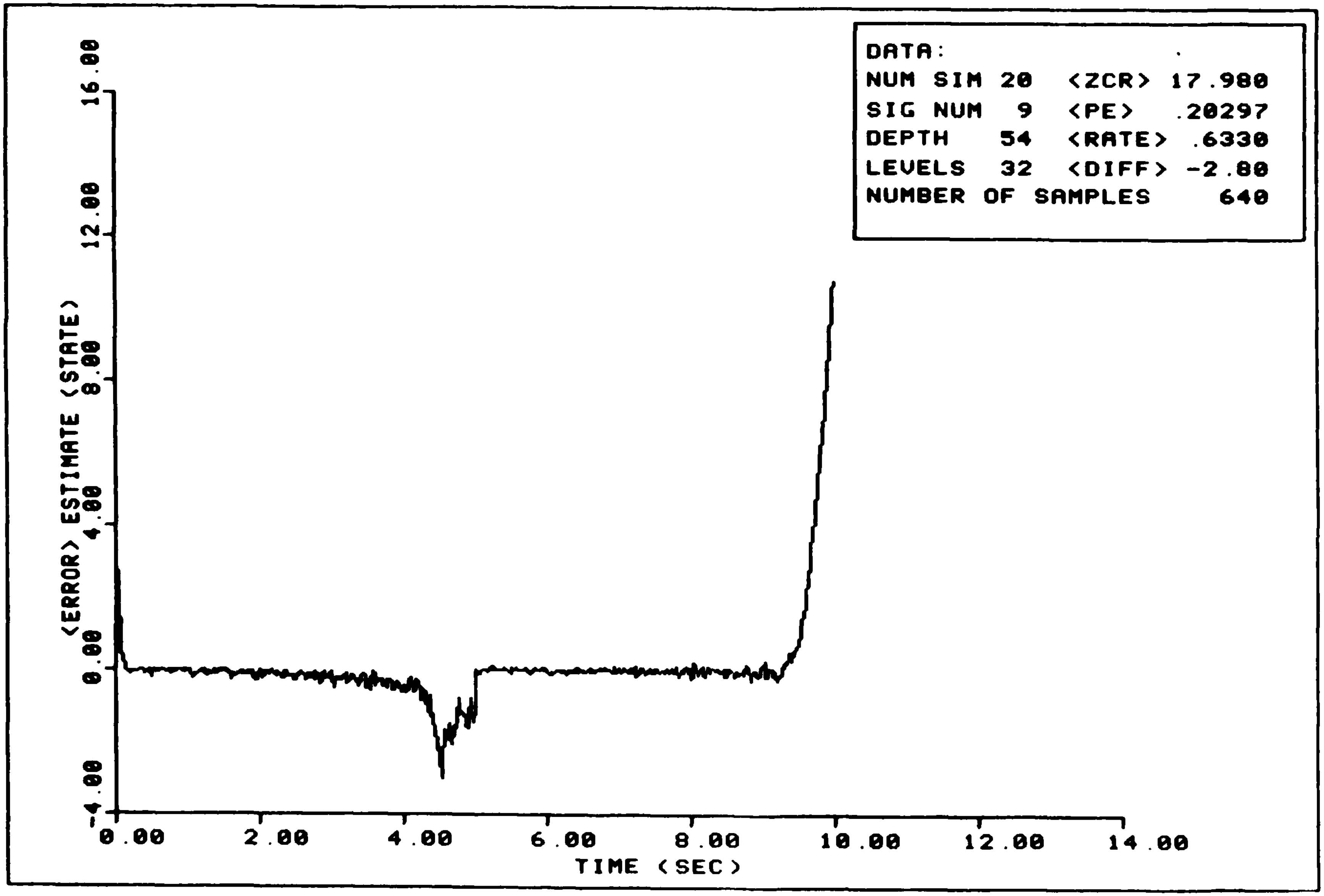


FIGURE 144. Ensemble Average State Error Performance

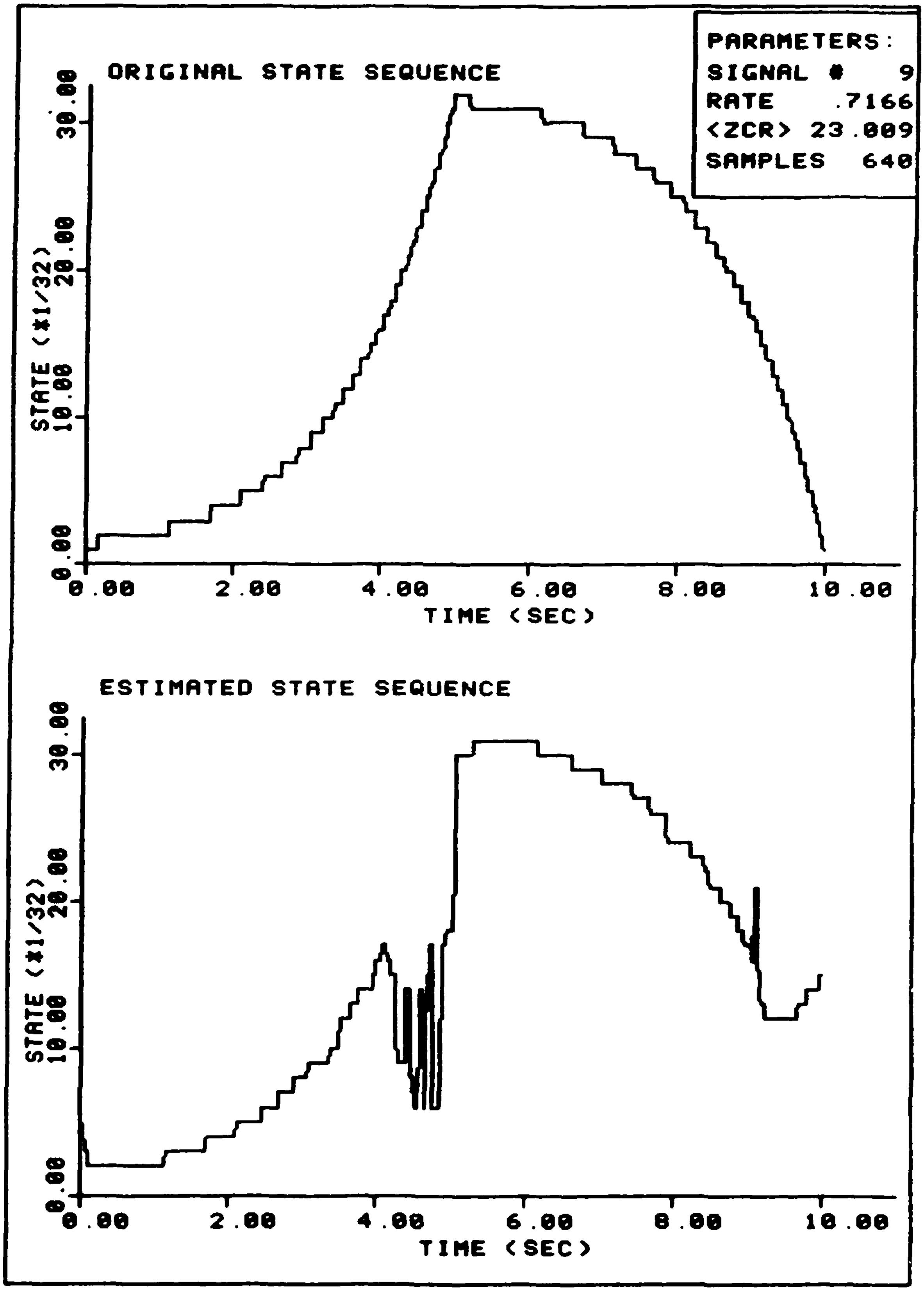


FIGURE 145. Sample Estimation

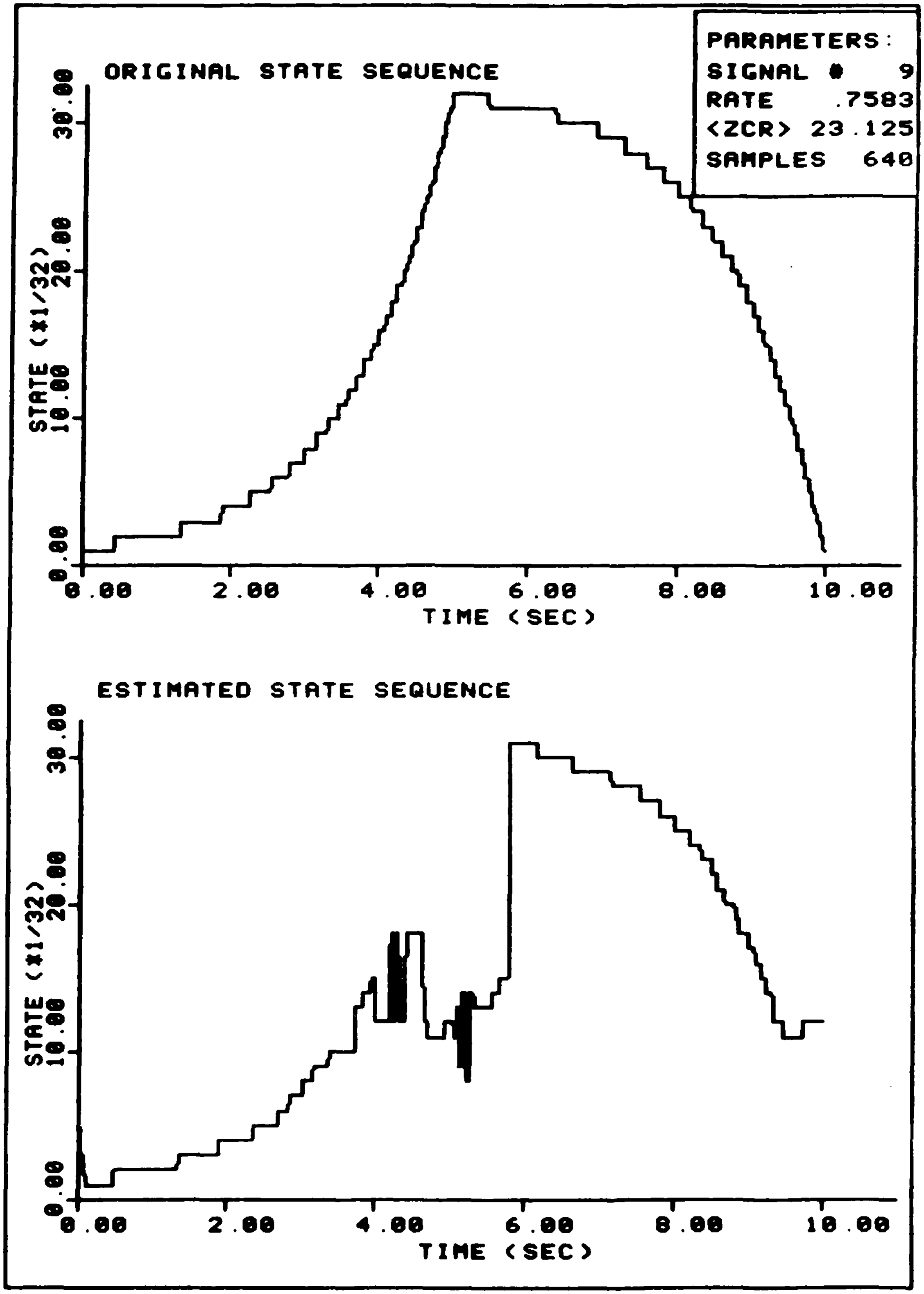


FIGURE 146. Sample Estimation

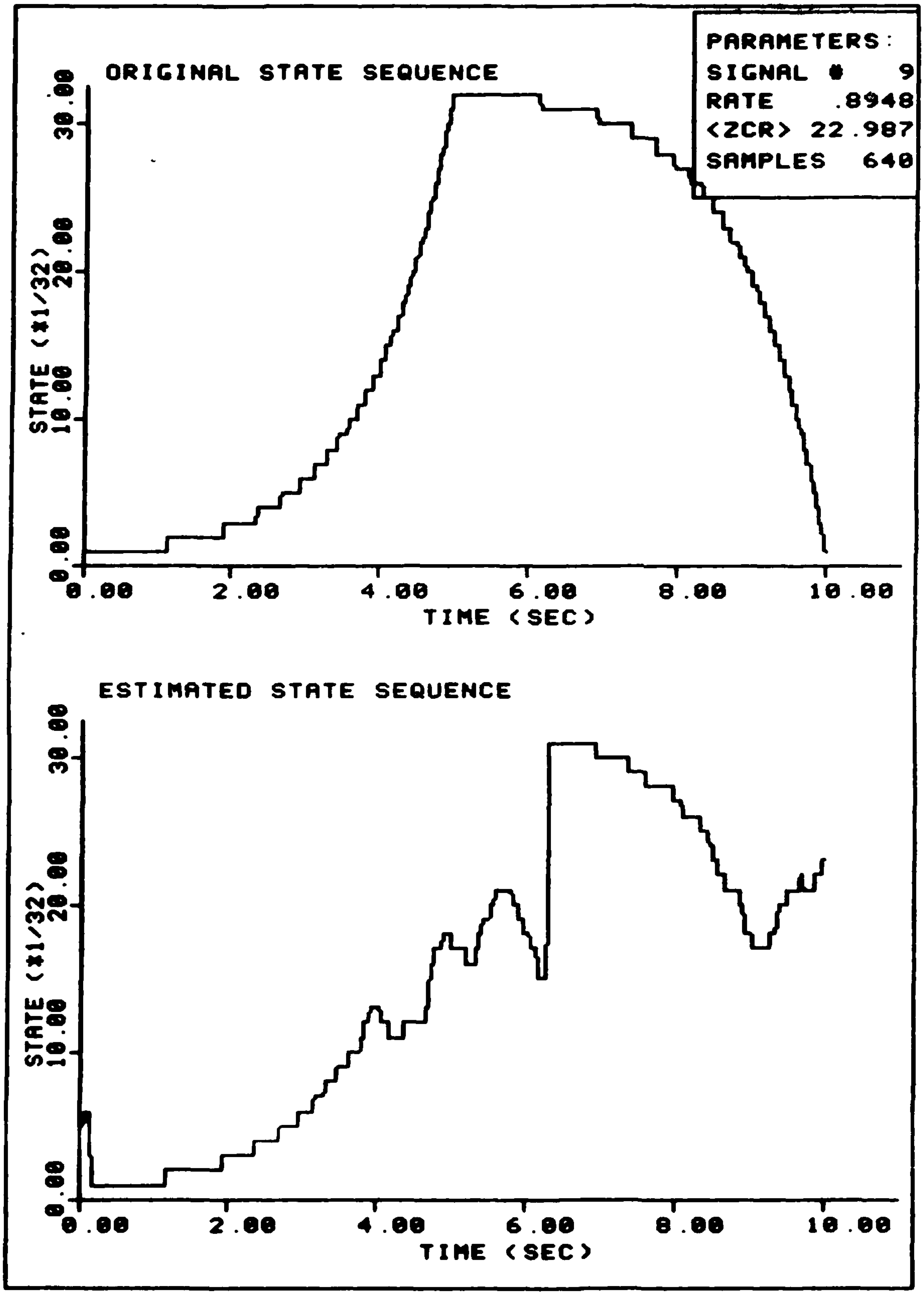


FIGURE 147. Sample Estimation

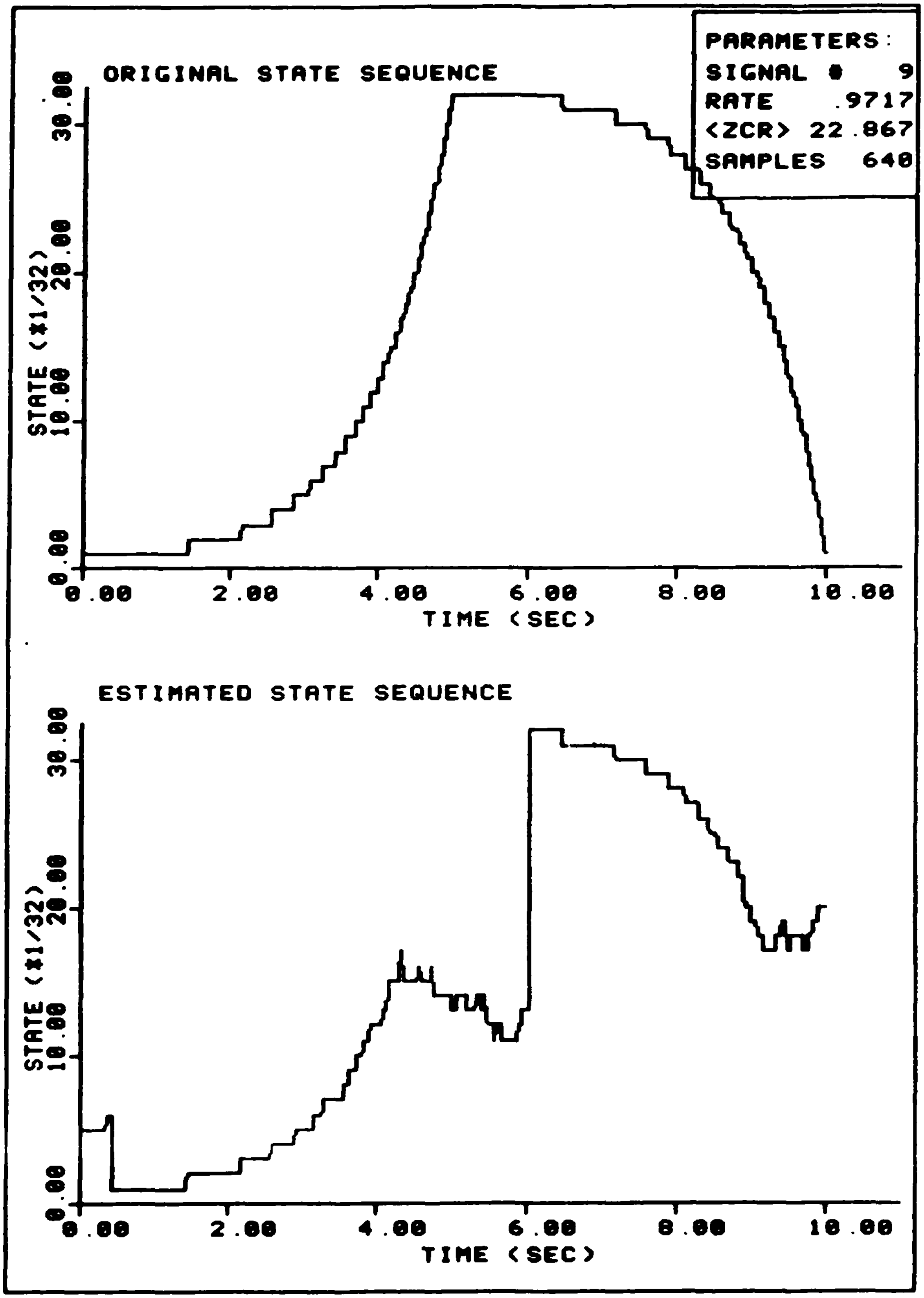


FIGURE 148. Sample Estimation

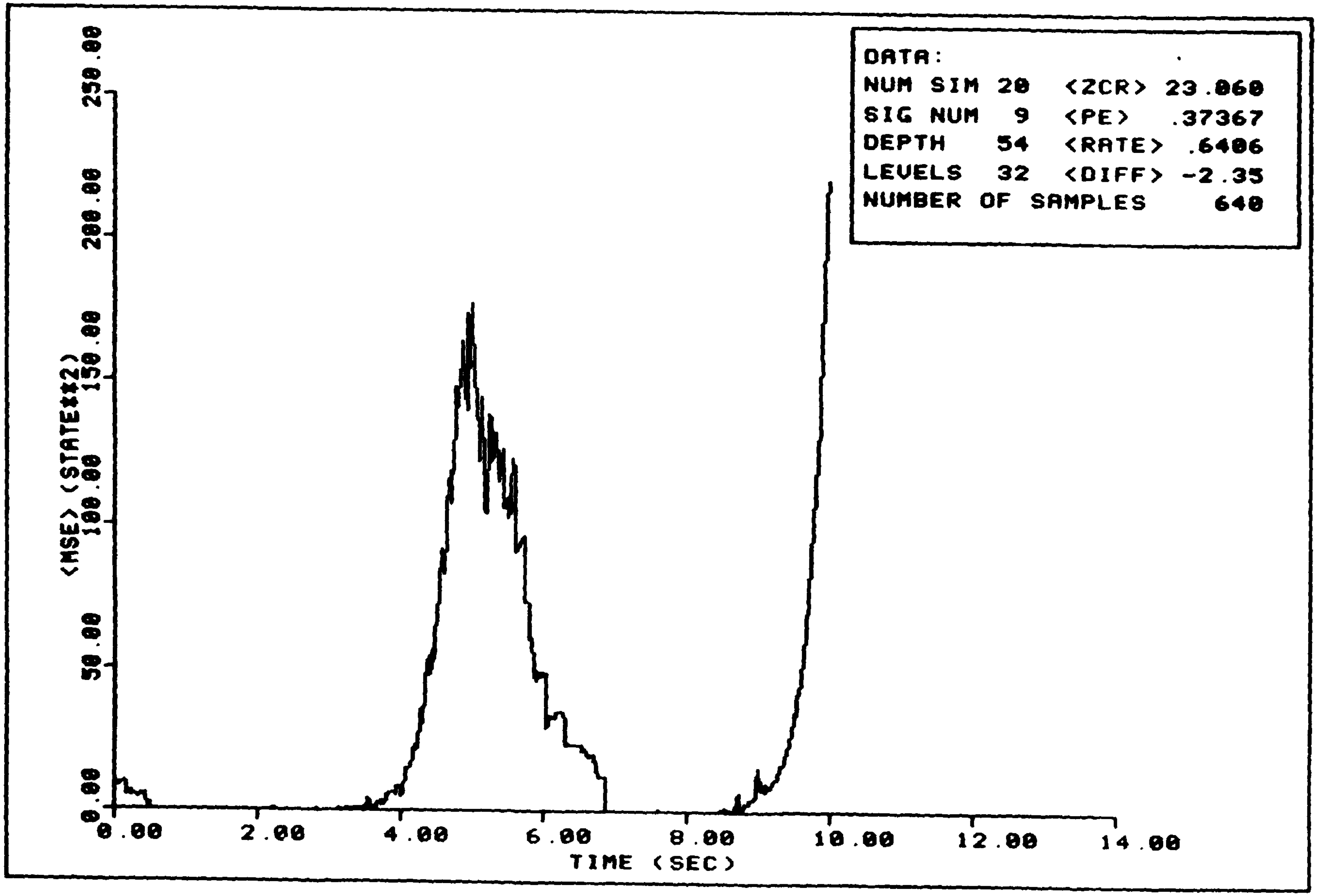


FIGURE 149. Ensemble Average MSE Performance

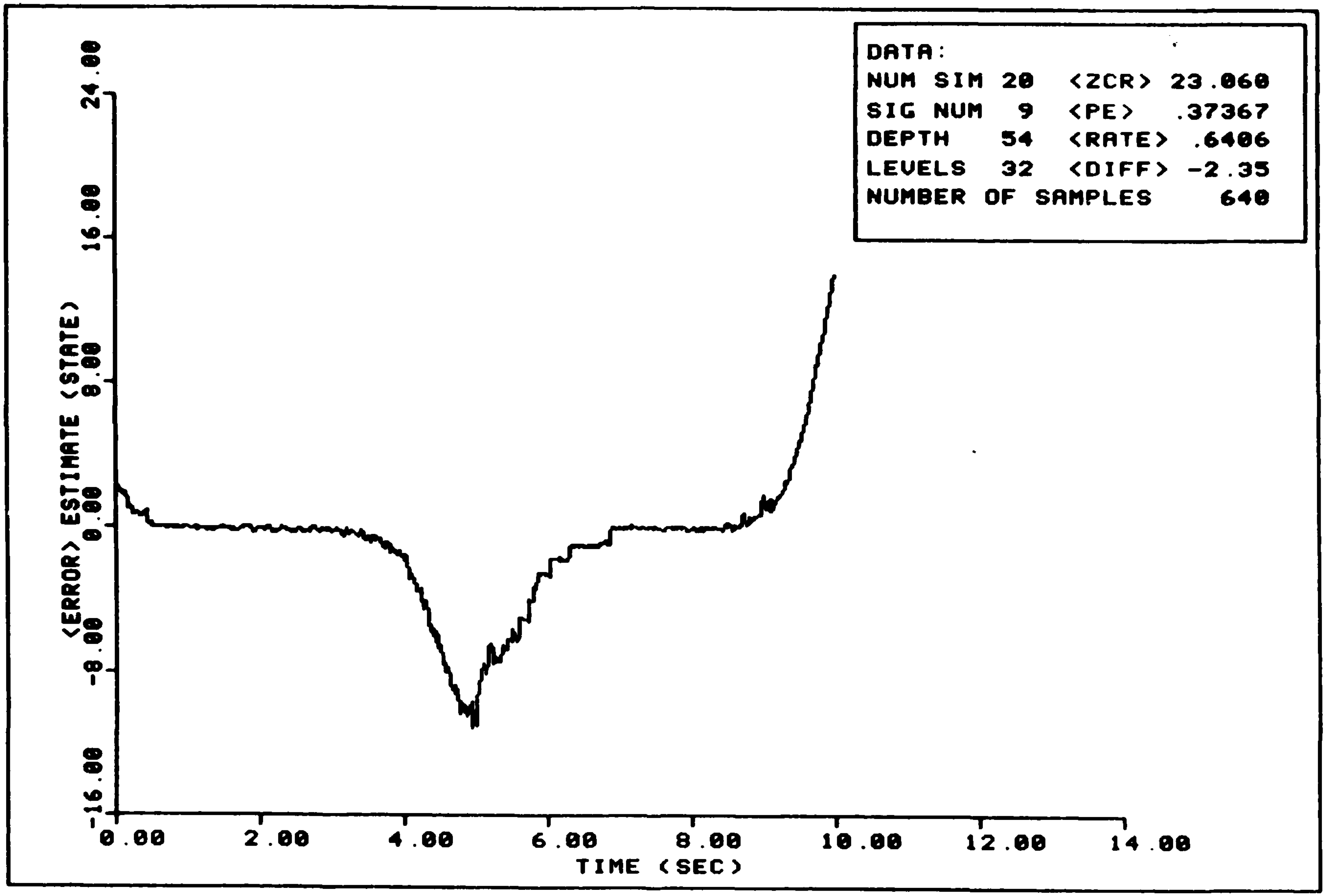


FIGURE 150. Ensemble Average State Error Performance

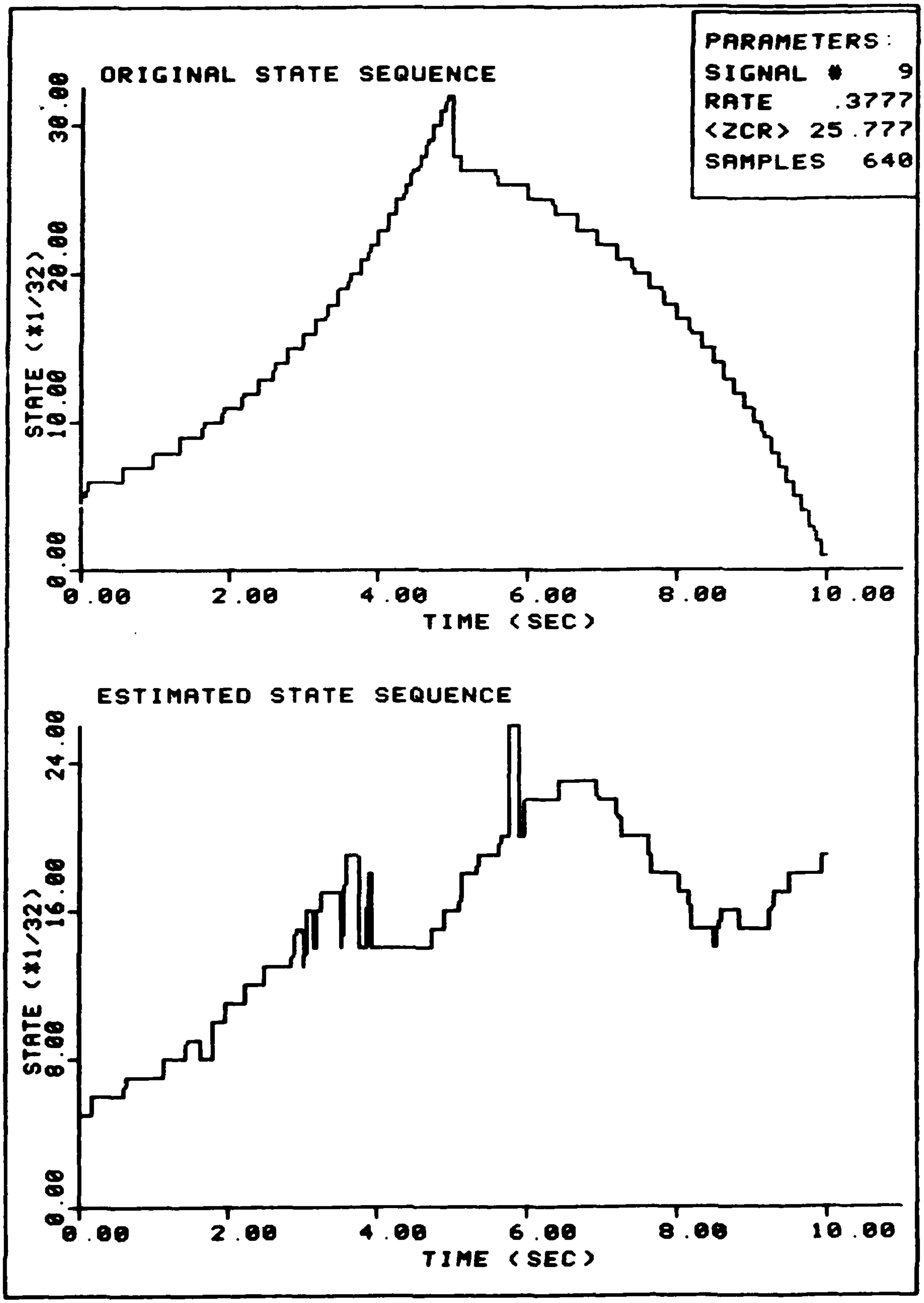


FIGURE 151. Sample Estimation

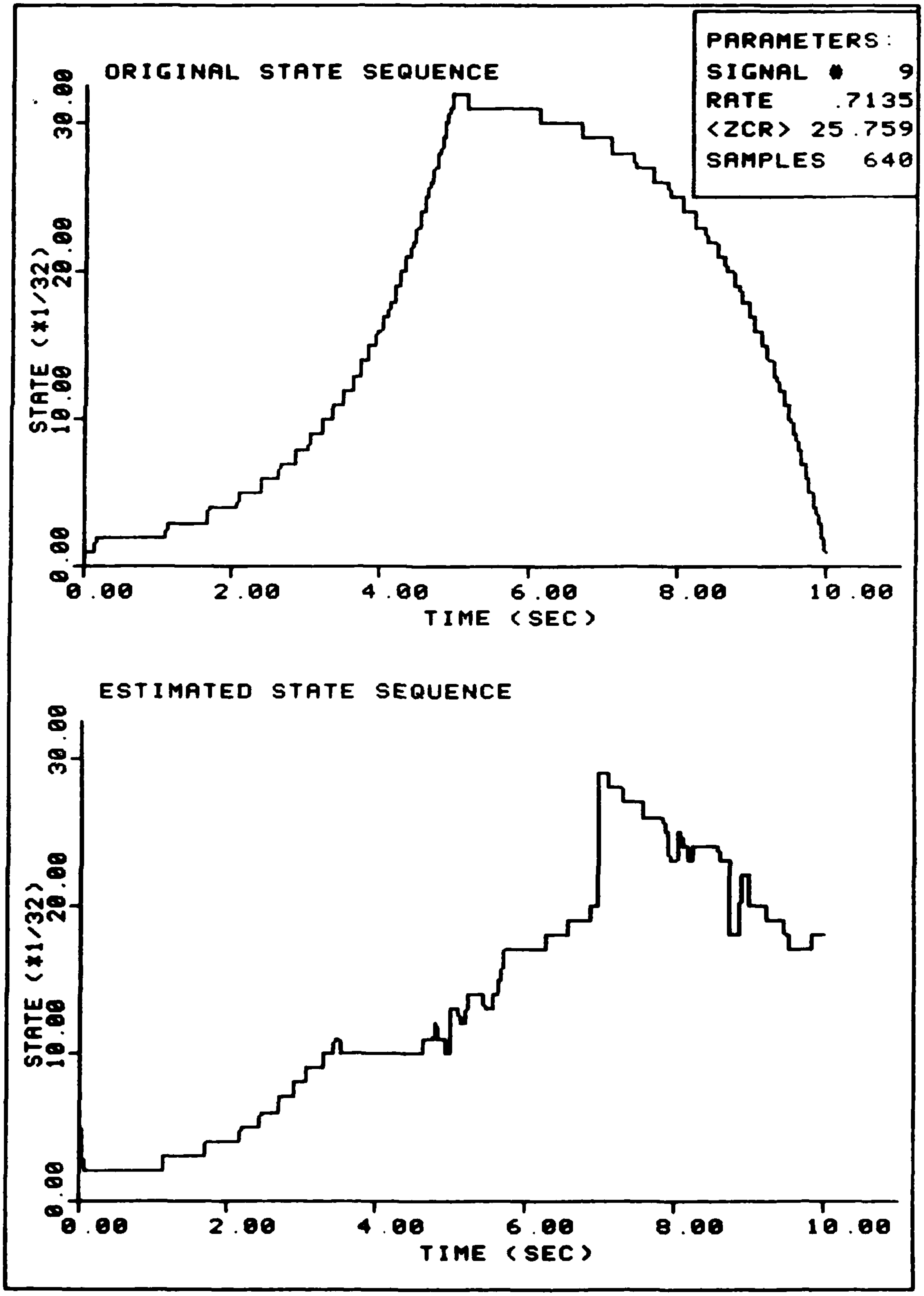


FIGURE 152. Sample Estimation

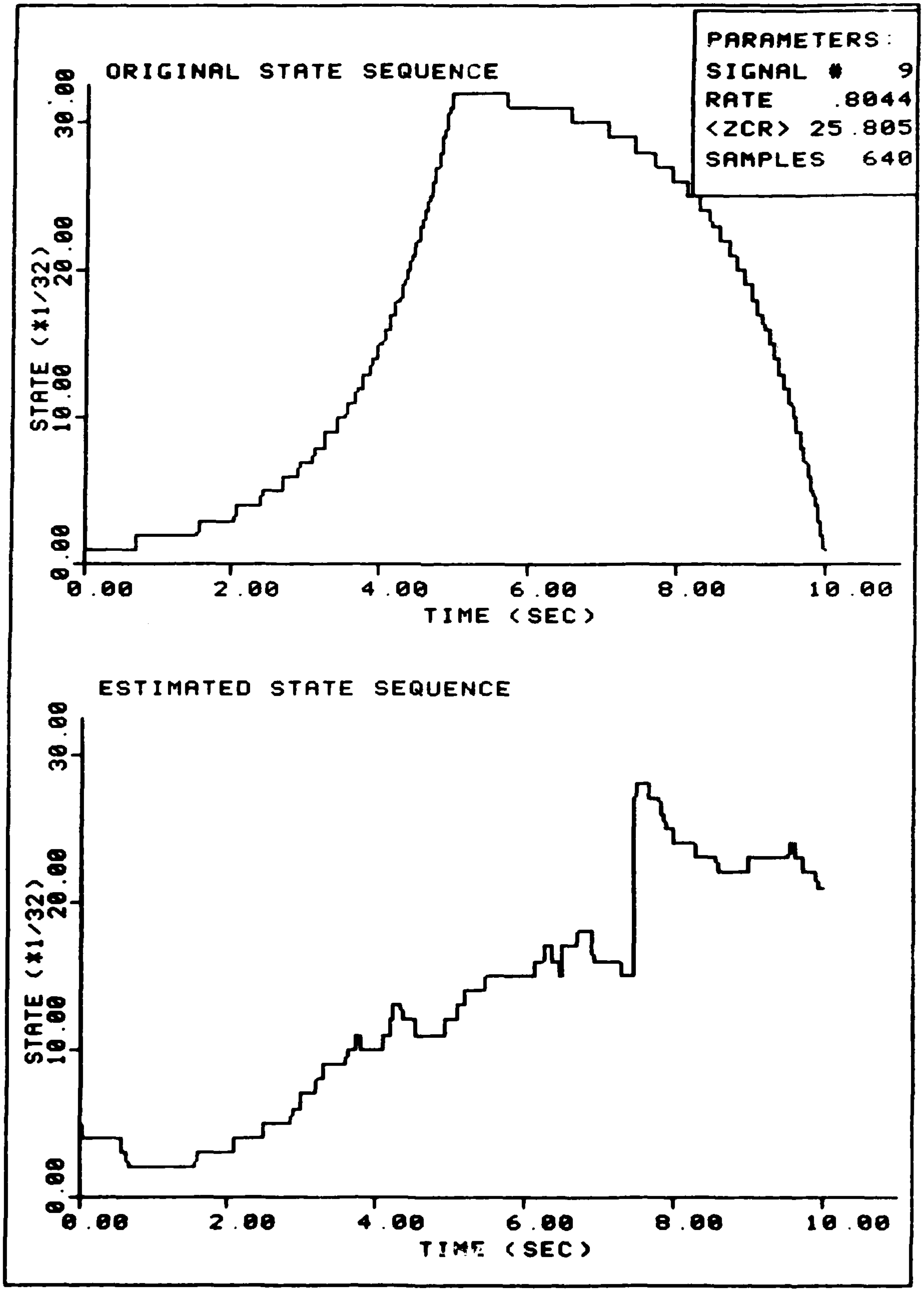


FIGURE 153. Sample Estimation

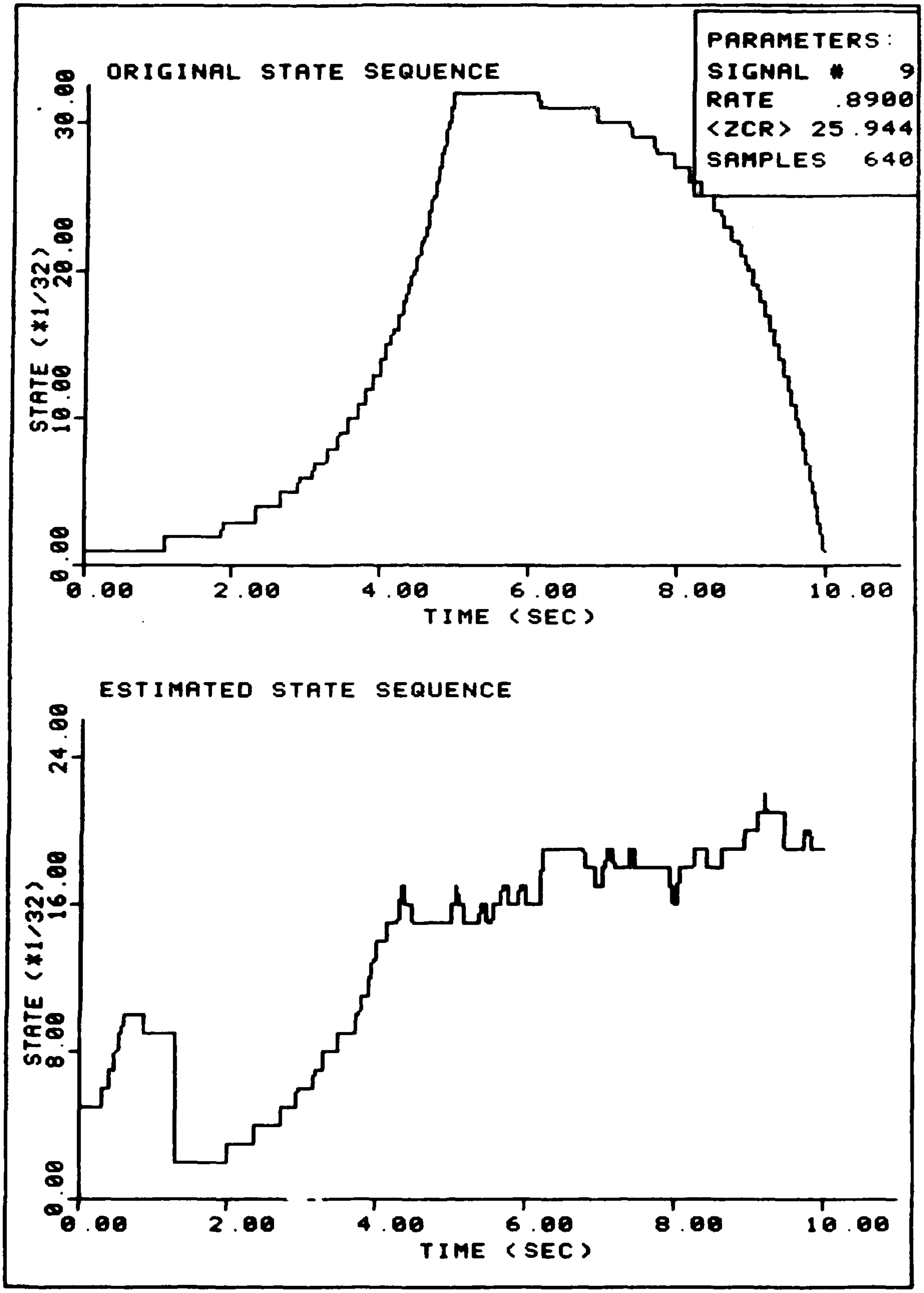


FIGURE 154. Sample Estimation

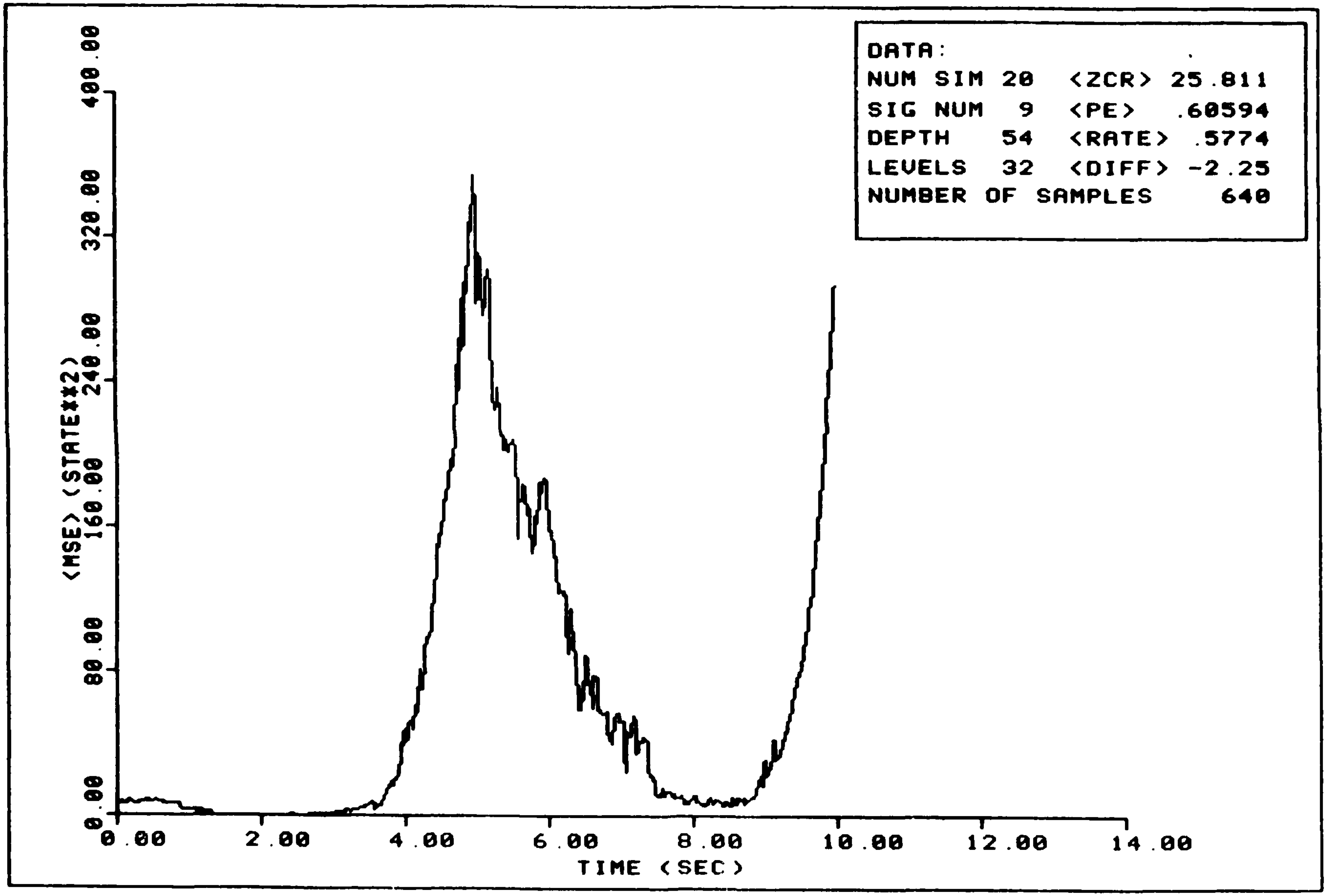


FIGURE 155. Ensemble Average MSE Performance

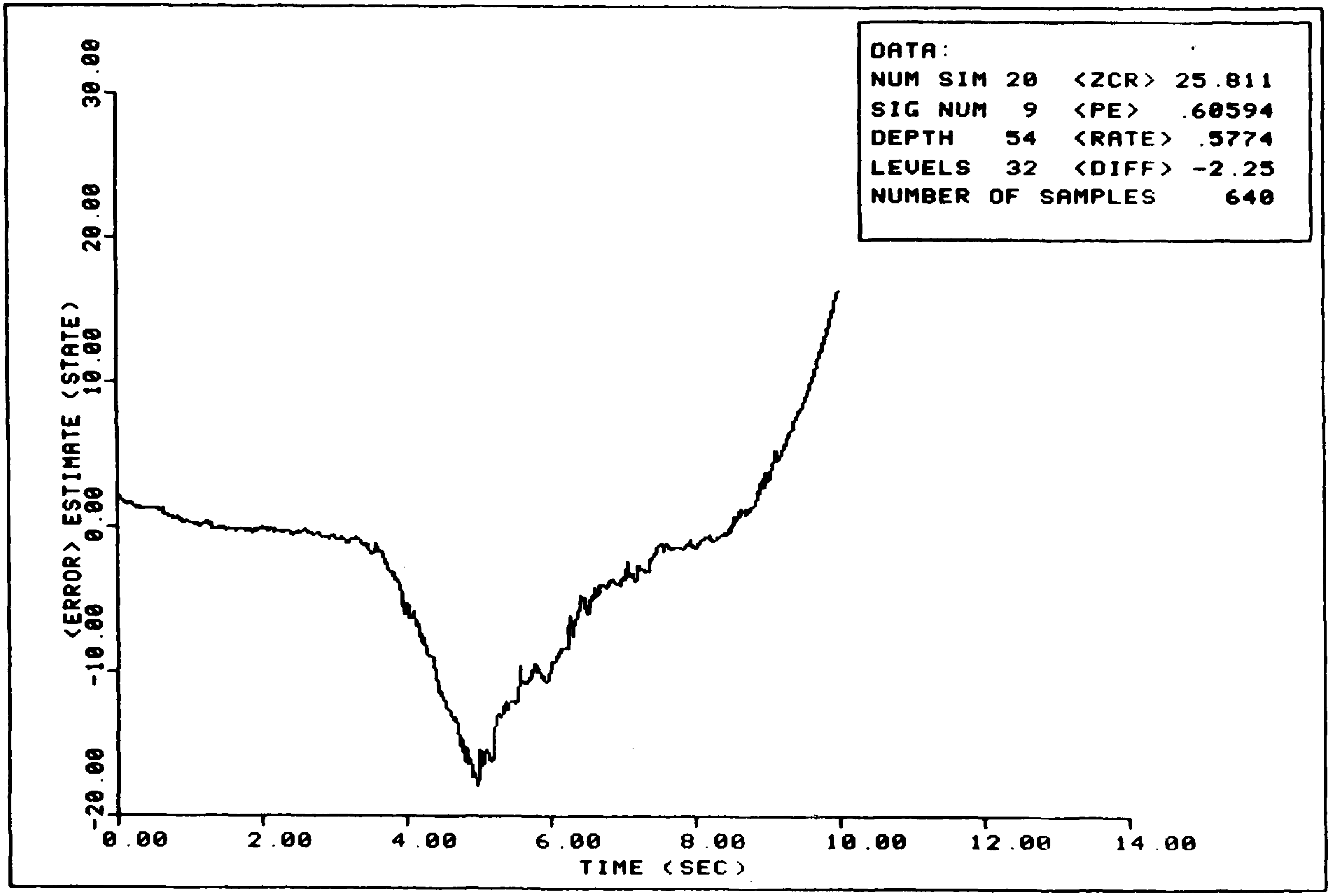


FIGURE 156. Ensemble Average State Error Performance

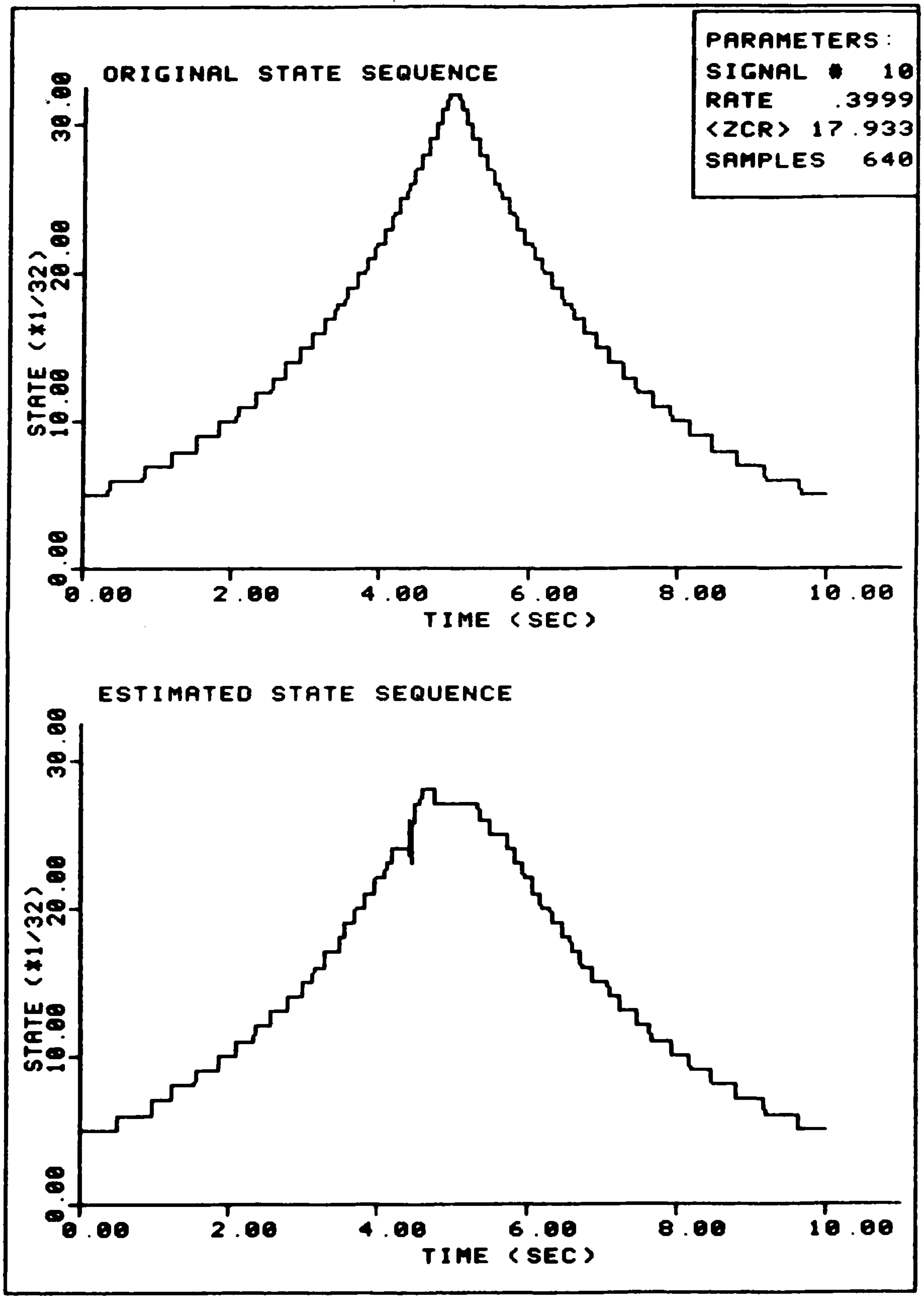


FIGURE 157. Sample Estimation

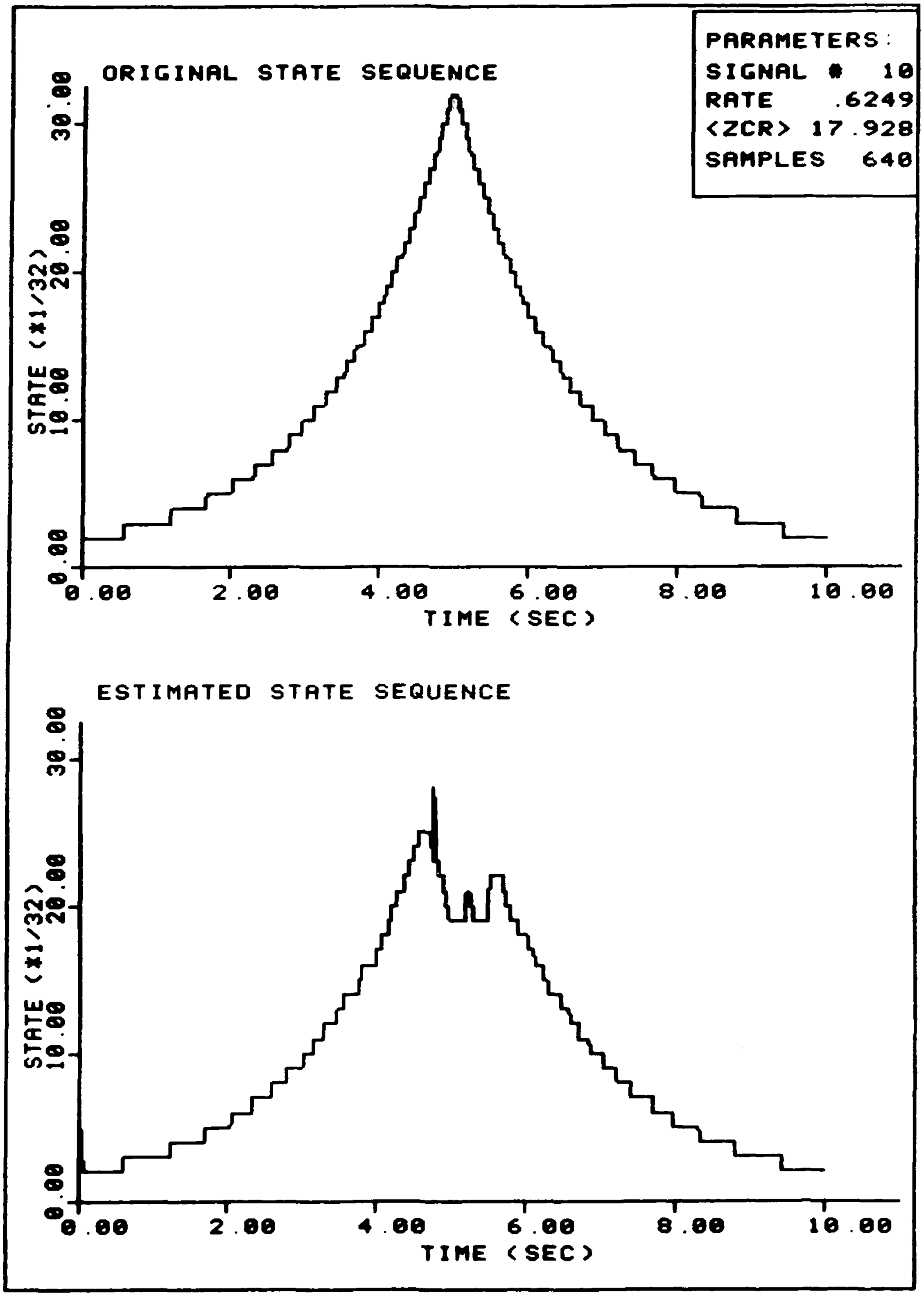


FIGURE 158. Sample Estimation

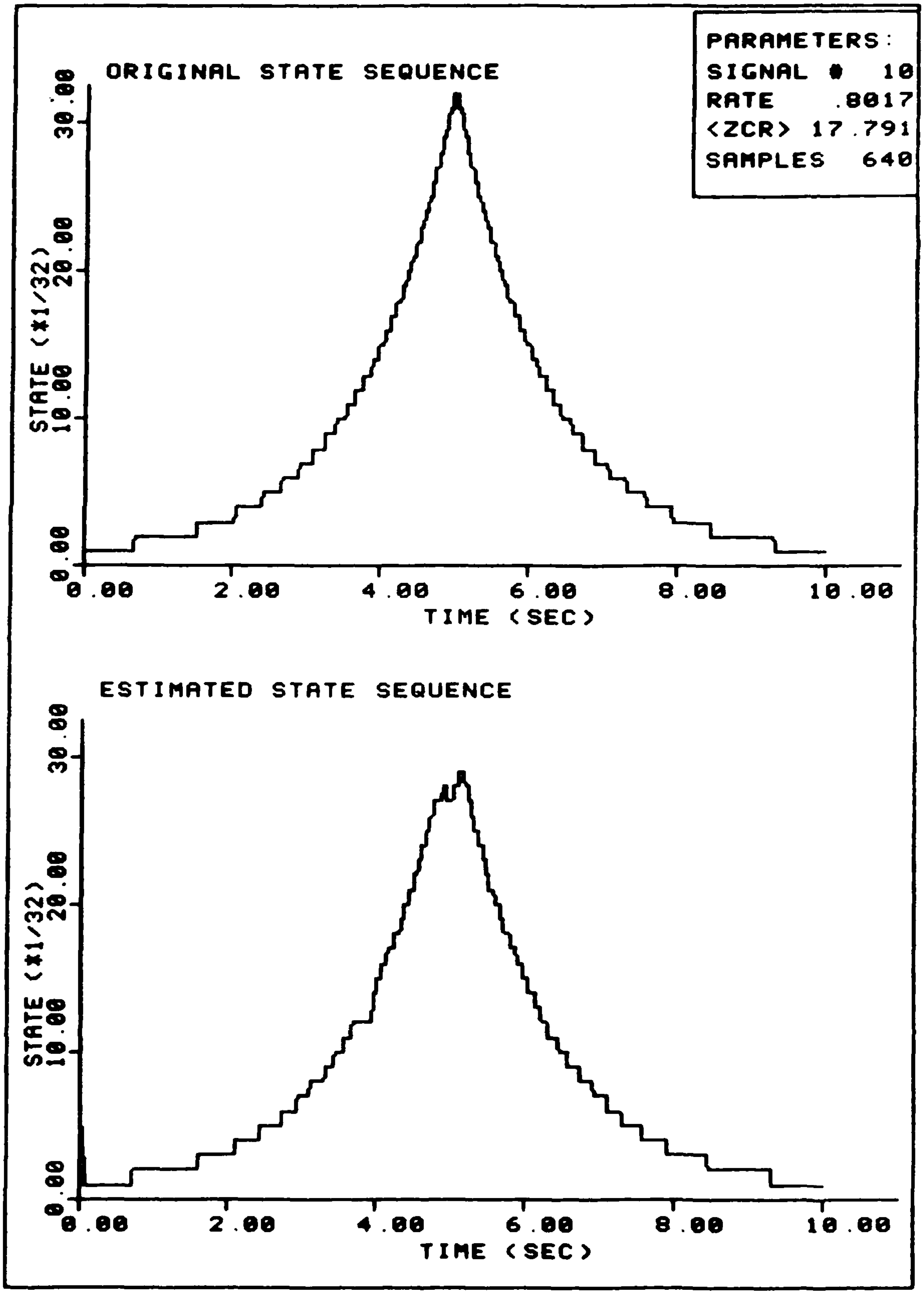


FIGURE 159. Sample Estimation

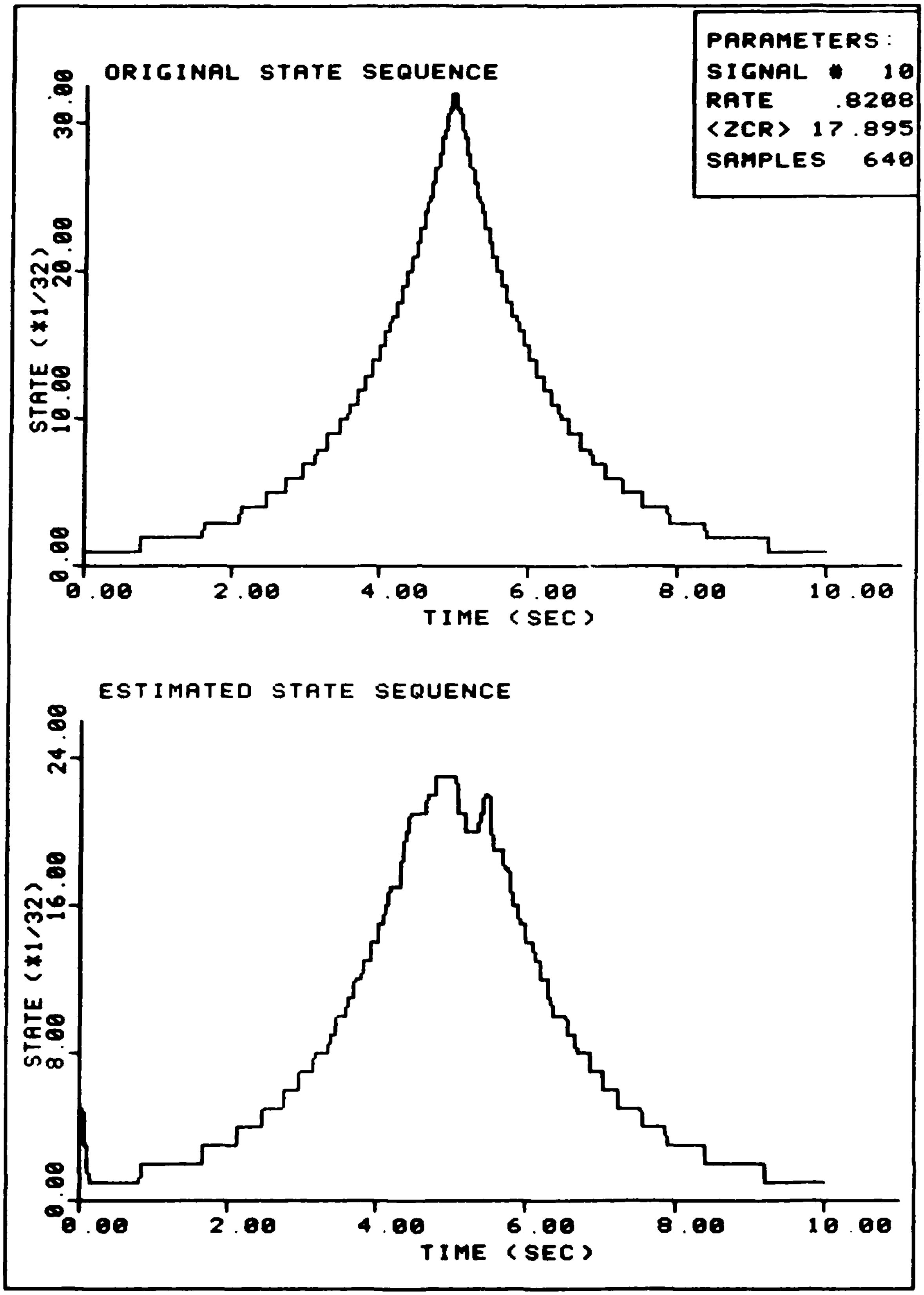


FIGURE 160. Sample Estimation

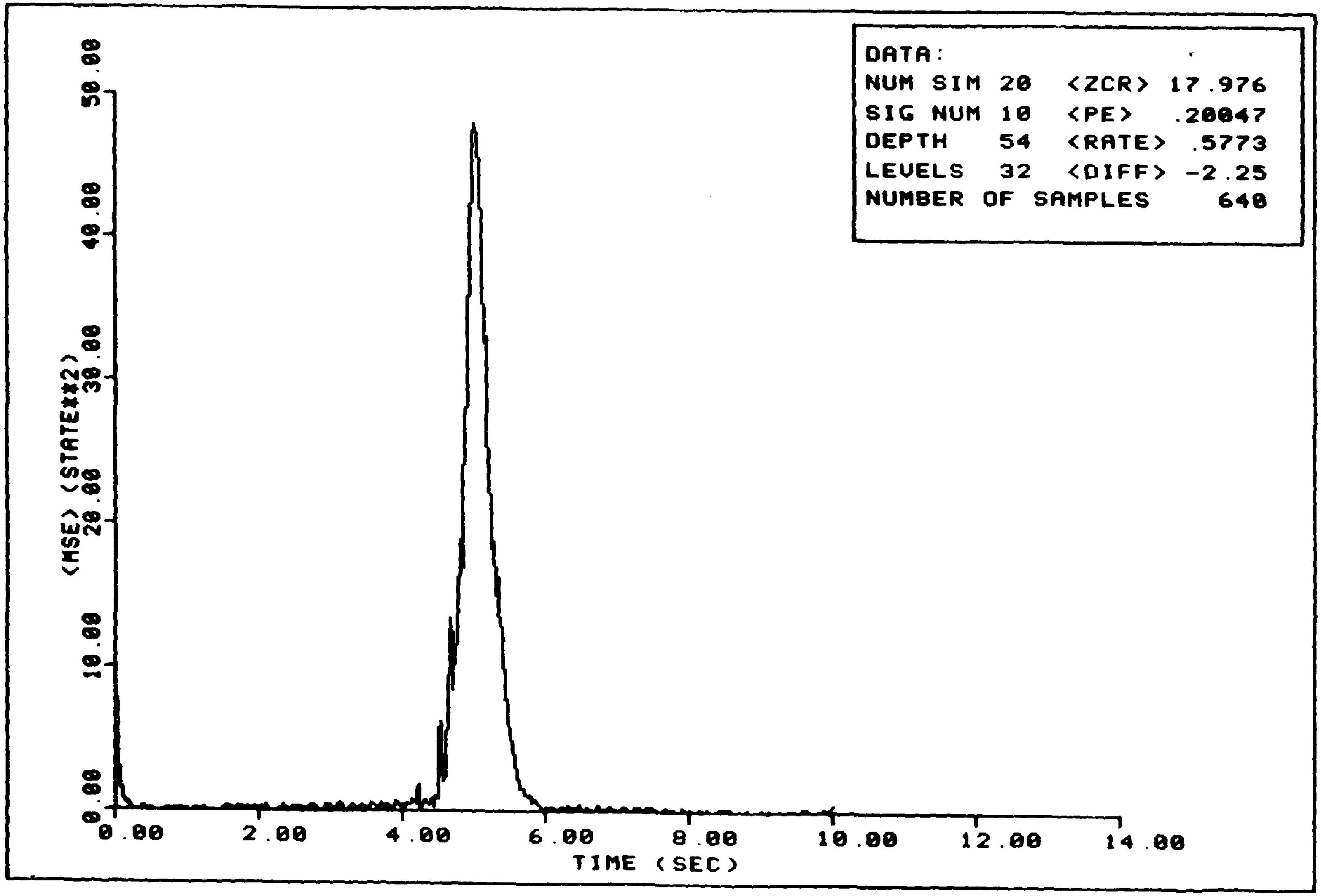


FIGURE 161. Ensemble Average MSE Performance

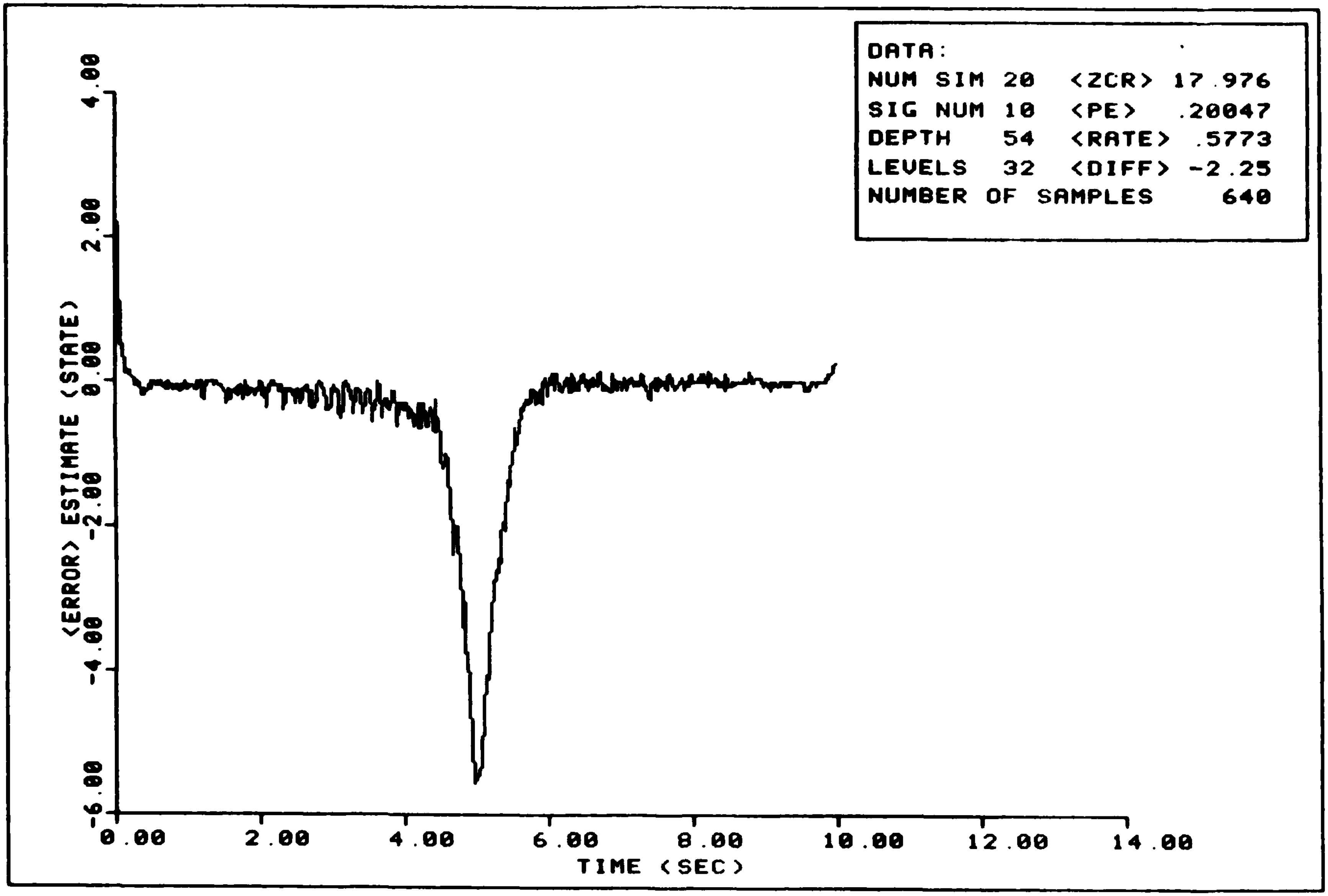


FIGURE 162. Ensemble Average State Error Performance

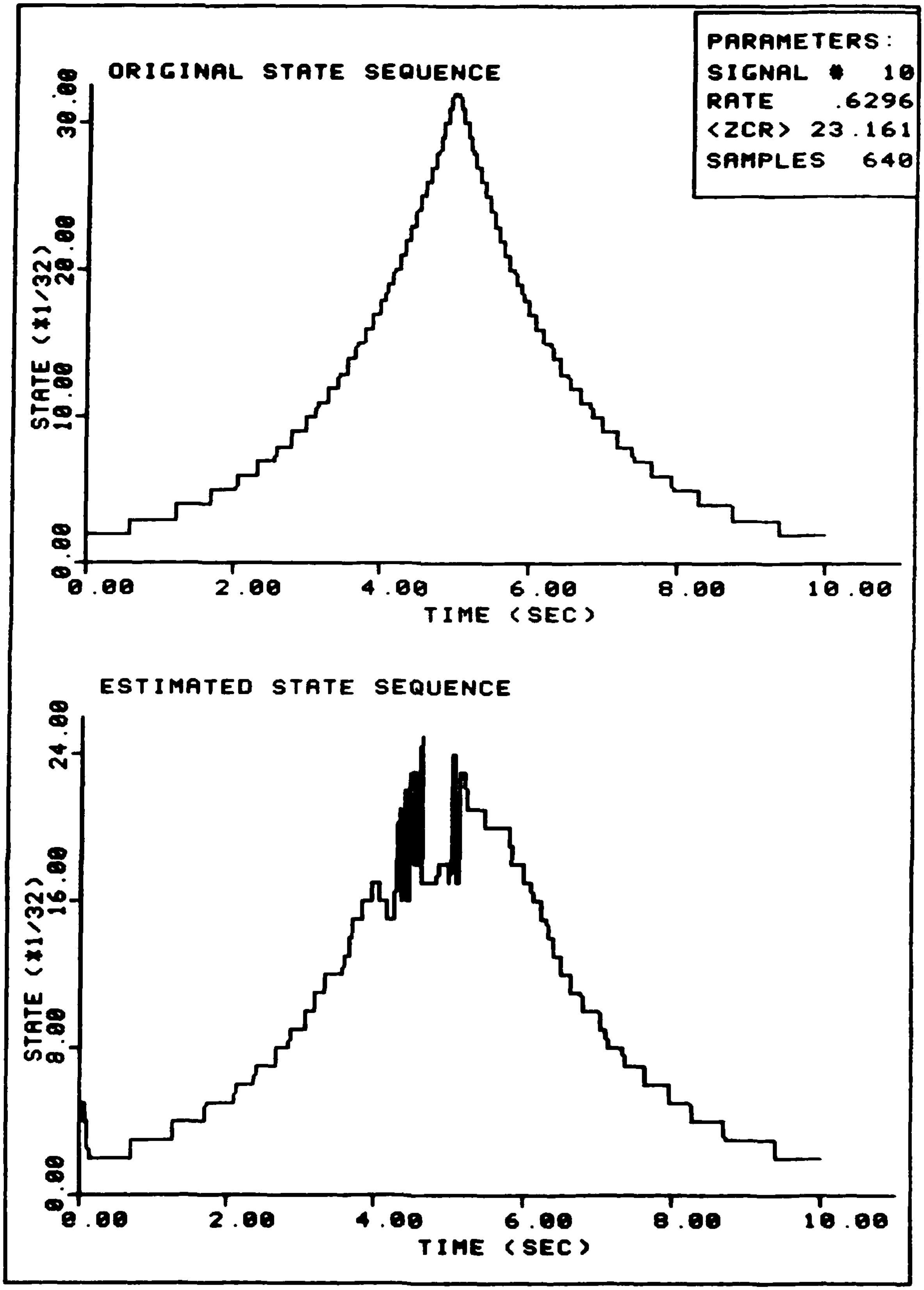


FIGURE 163. Sample Estimation

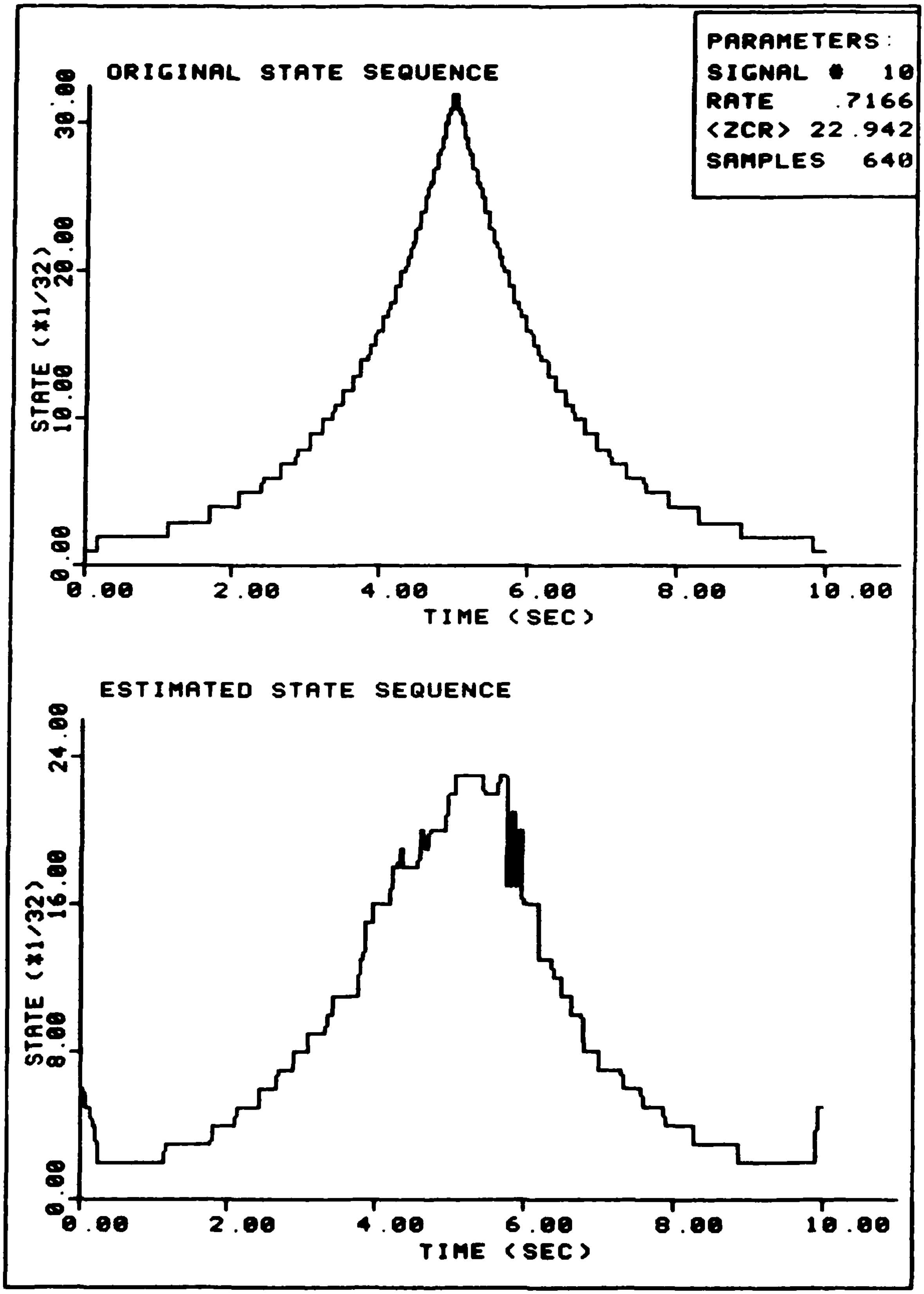


FIGURE 164. Sample Estimation

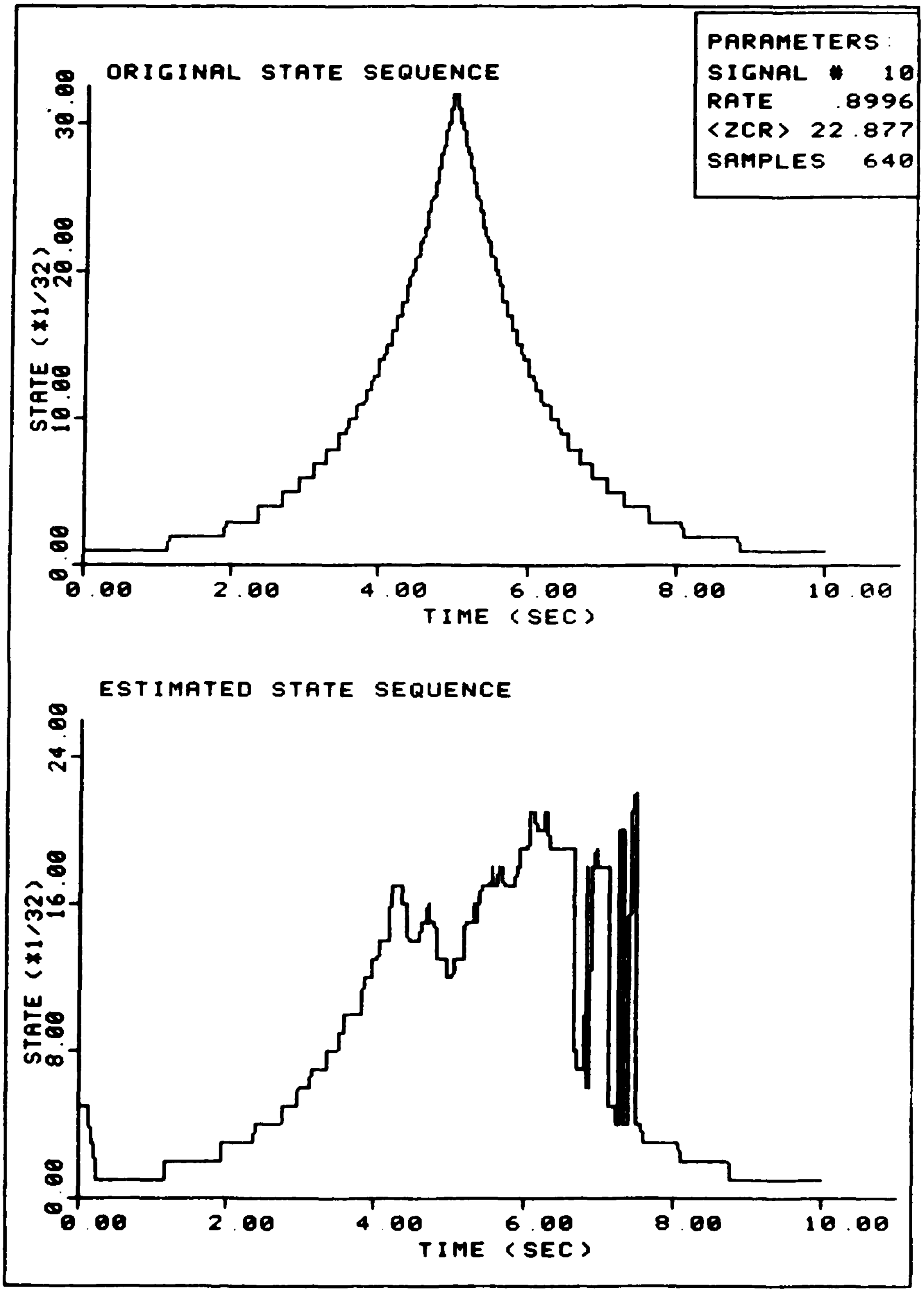


FIGURE 165. Sample Estimation

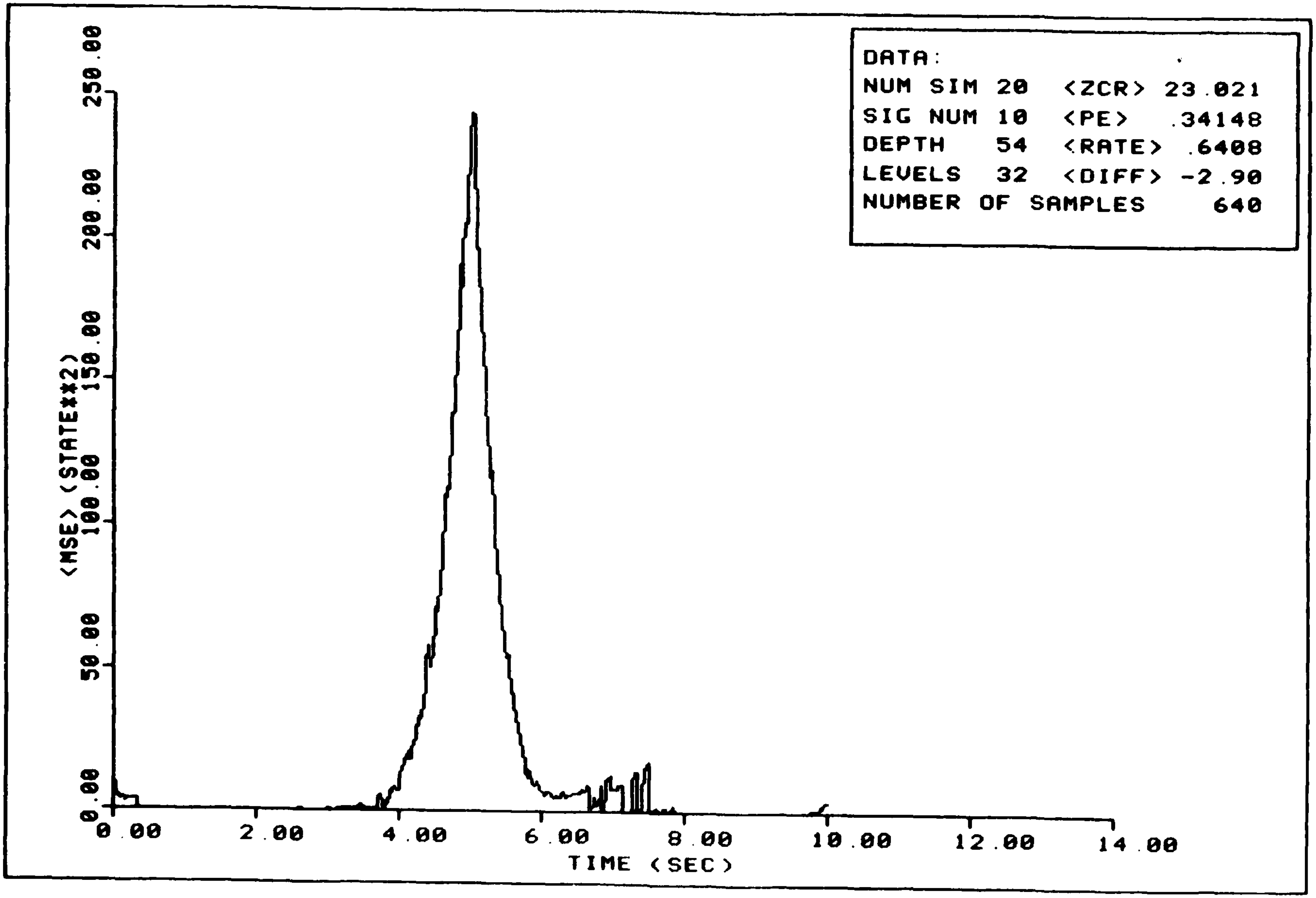


FIGURE 166. Ensemble Average MSE Performance

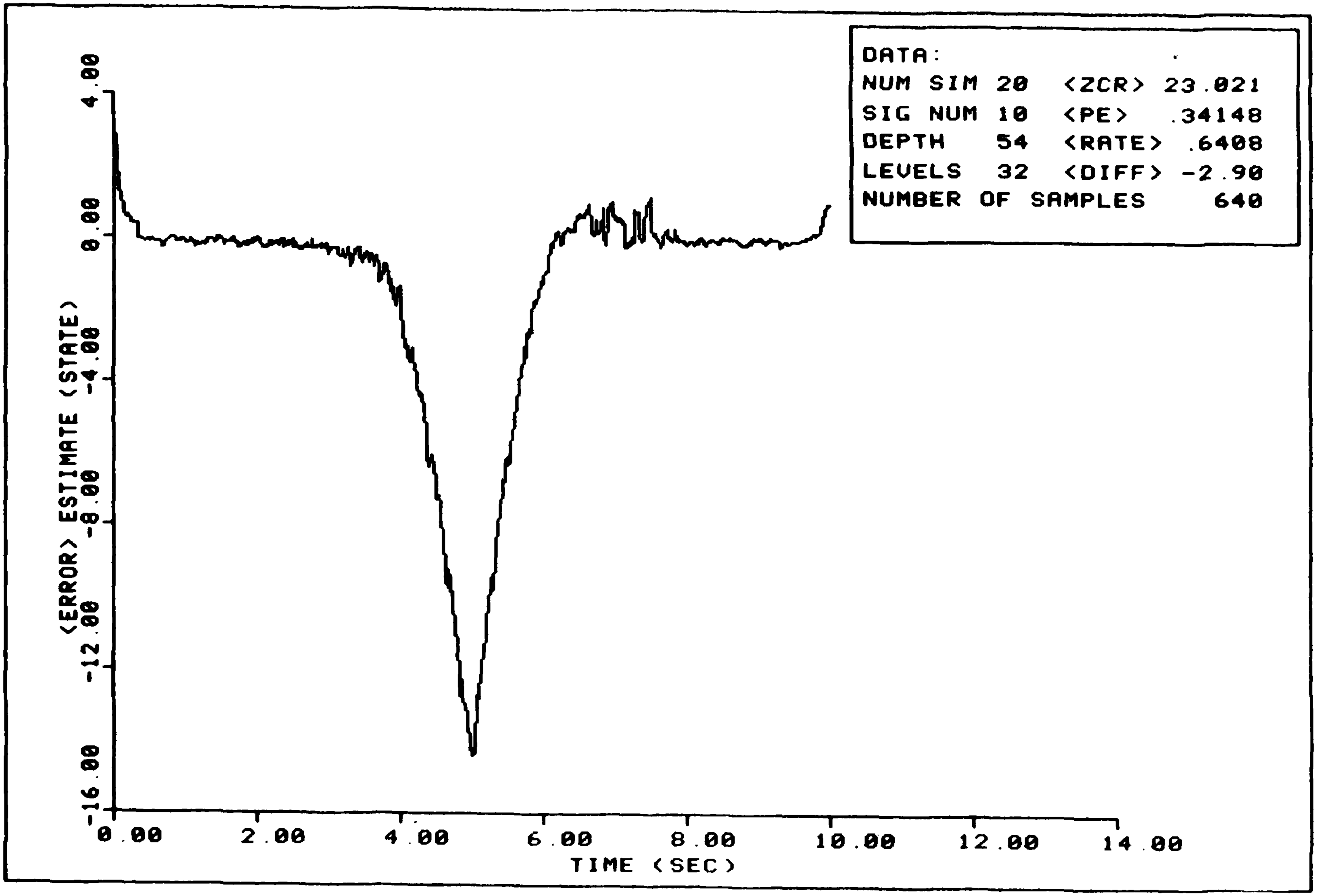


FIGURE 167. Ensemble Average State Error Performance

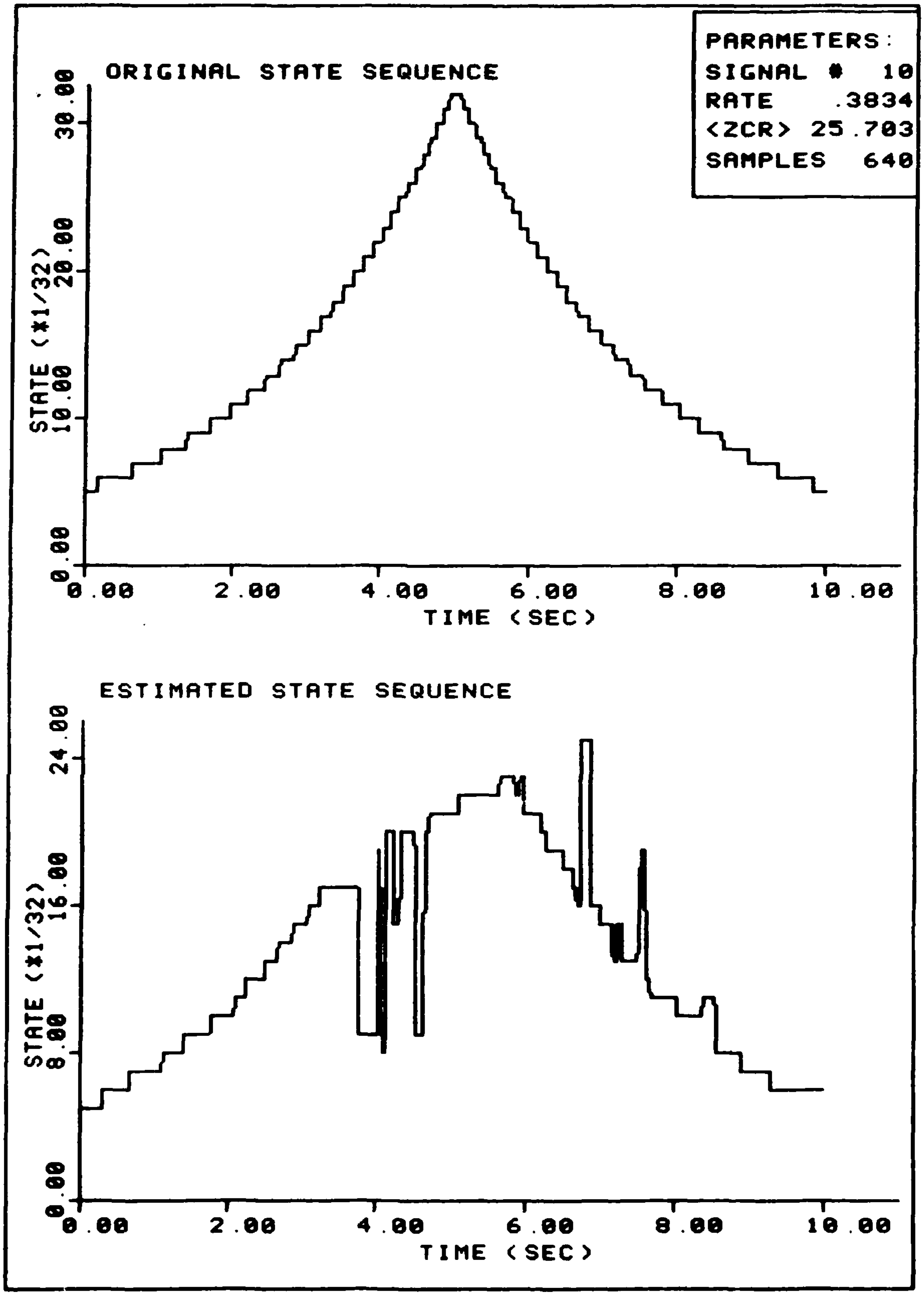


FIGURE 168. Sample Estimation

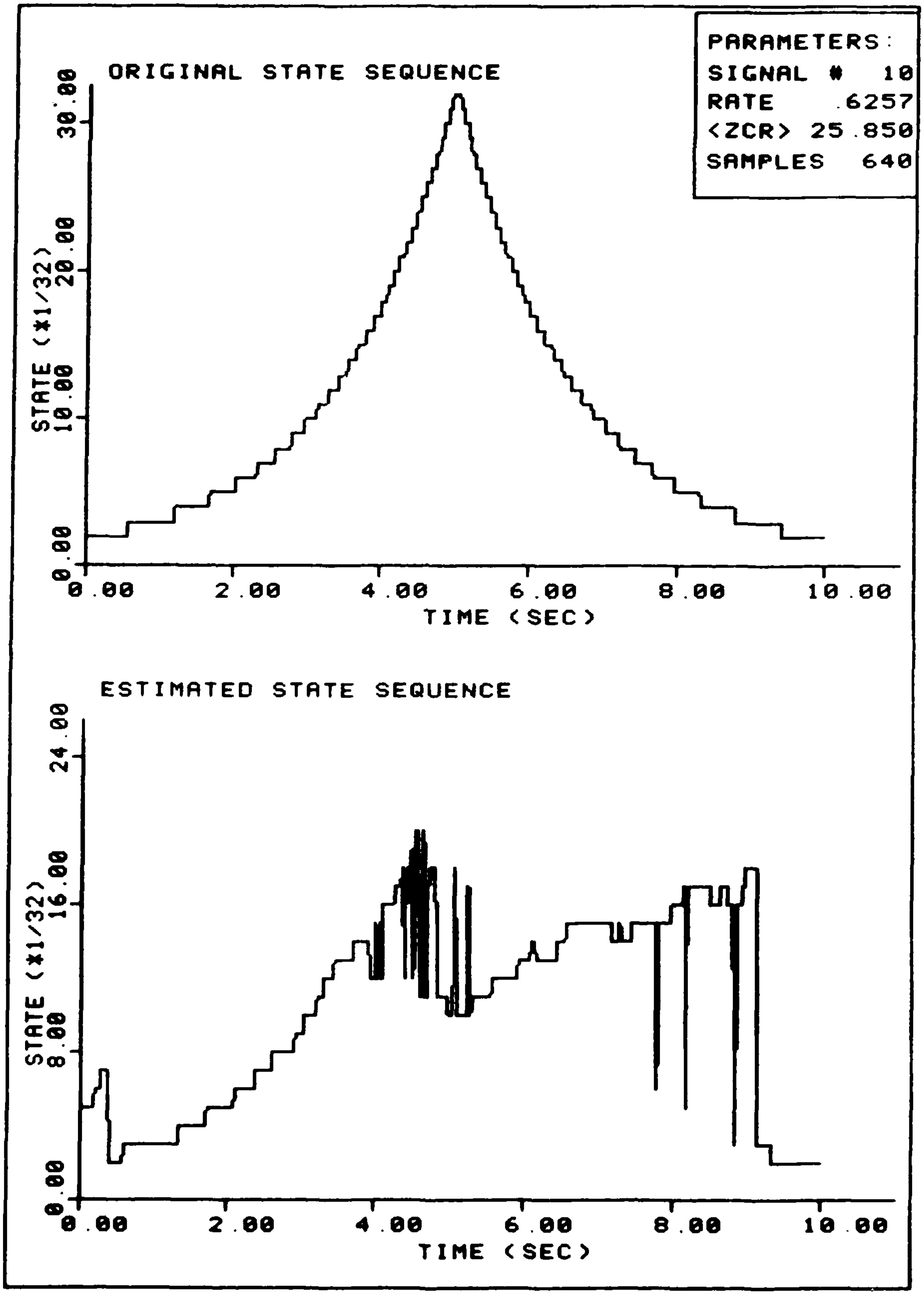


FIGURE 169. Sample Estimation

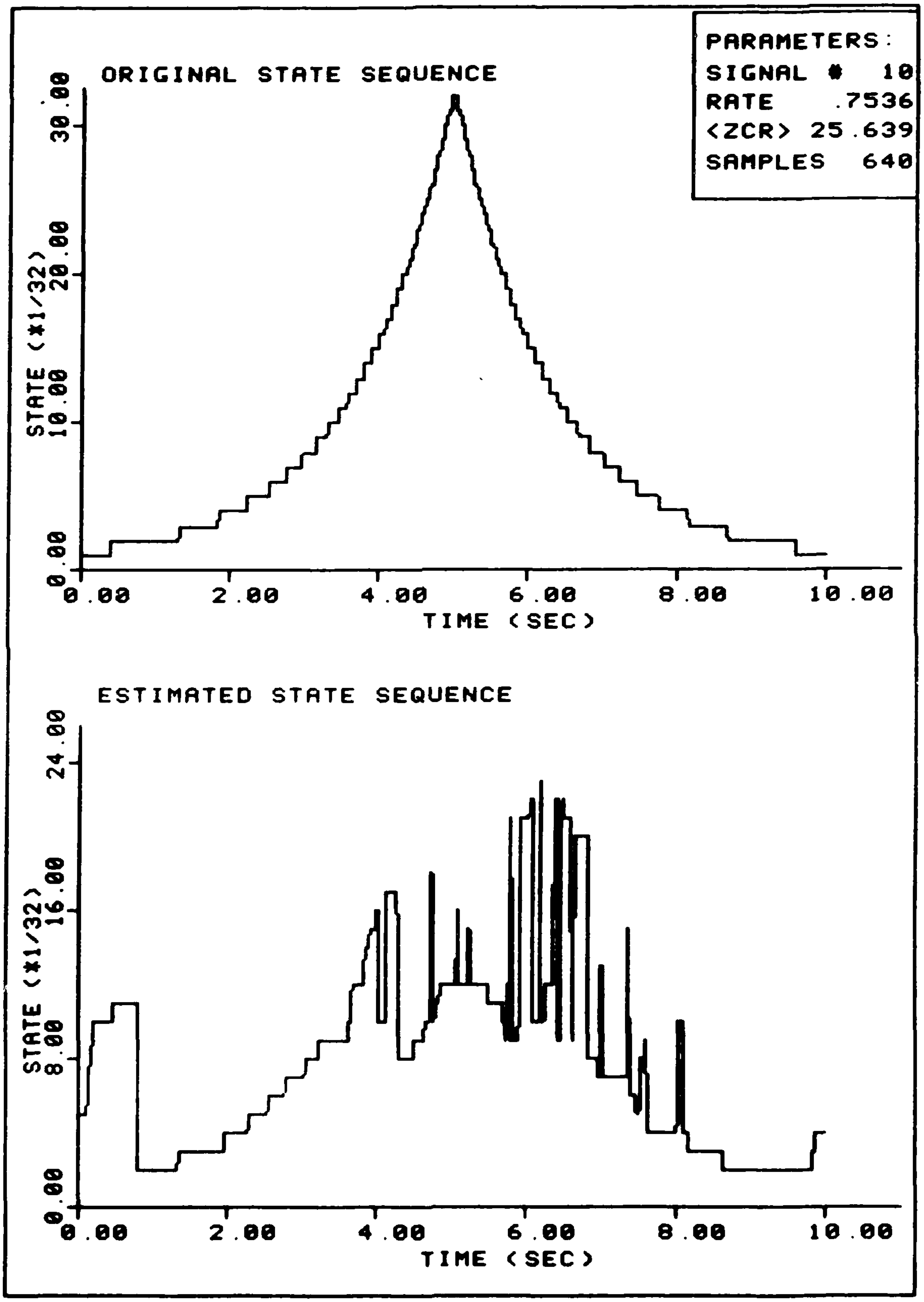


FIGURE 170. Sample Estimation

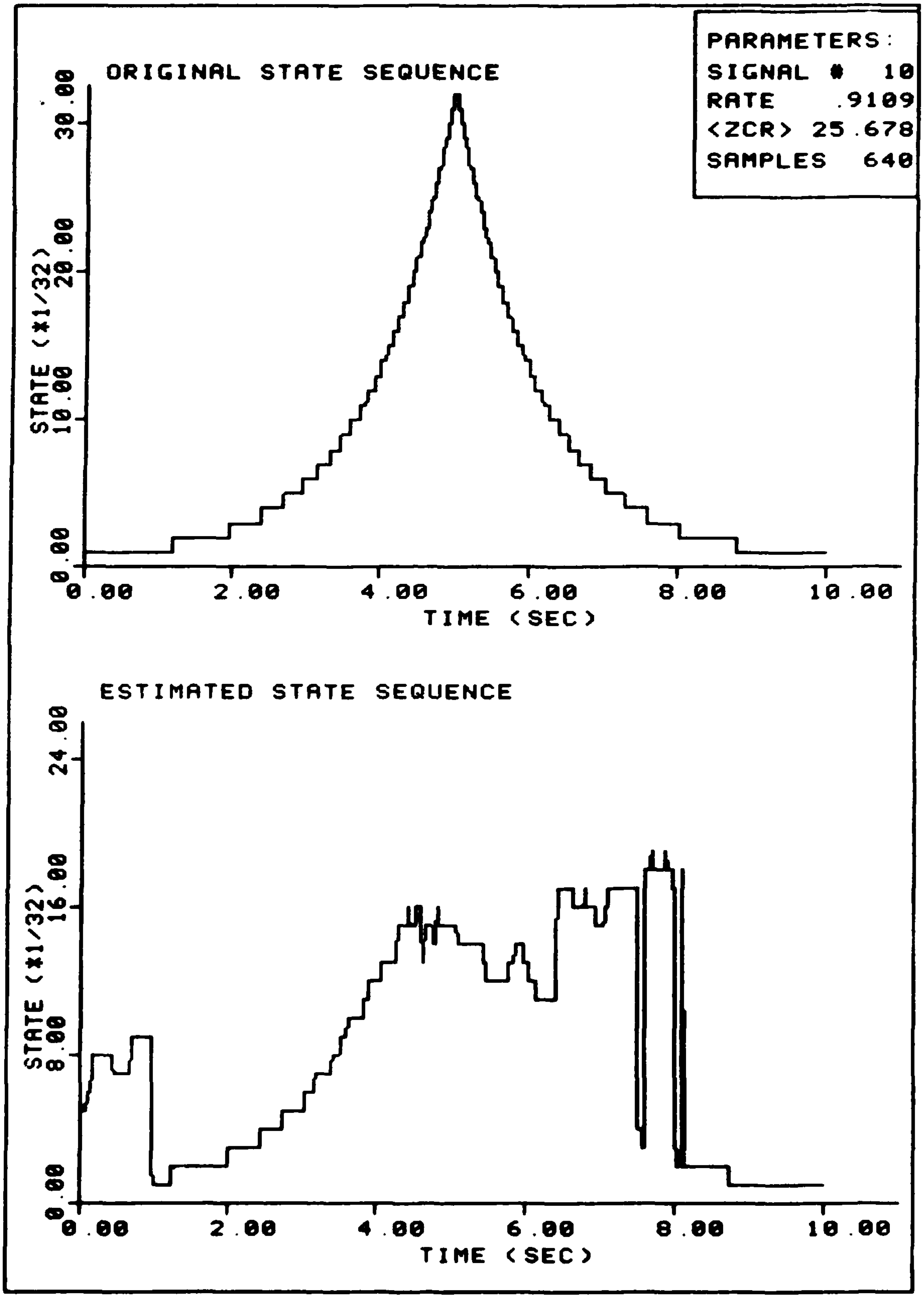


FIGURE 171. Sample Estimation

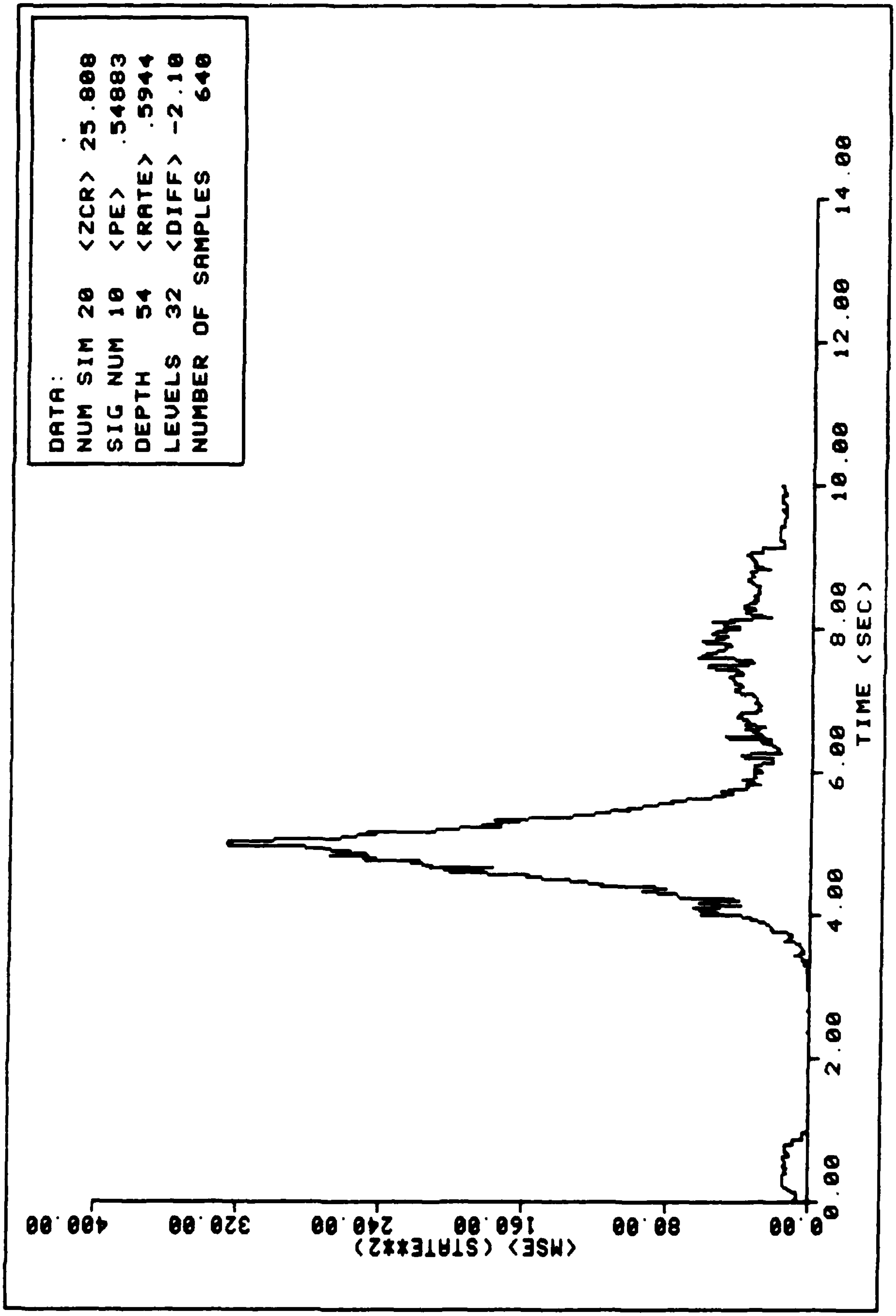


FIGURE 172. Ensemble Average MSE Performance

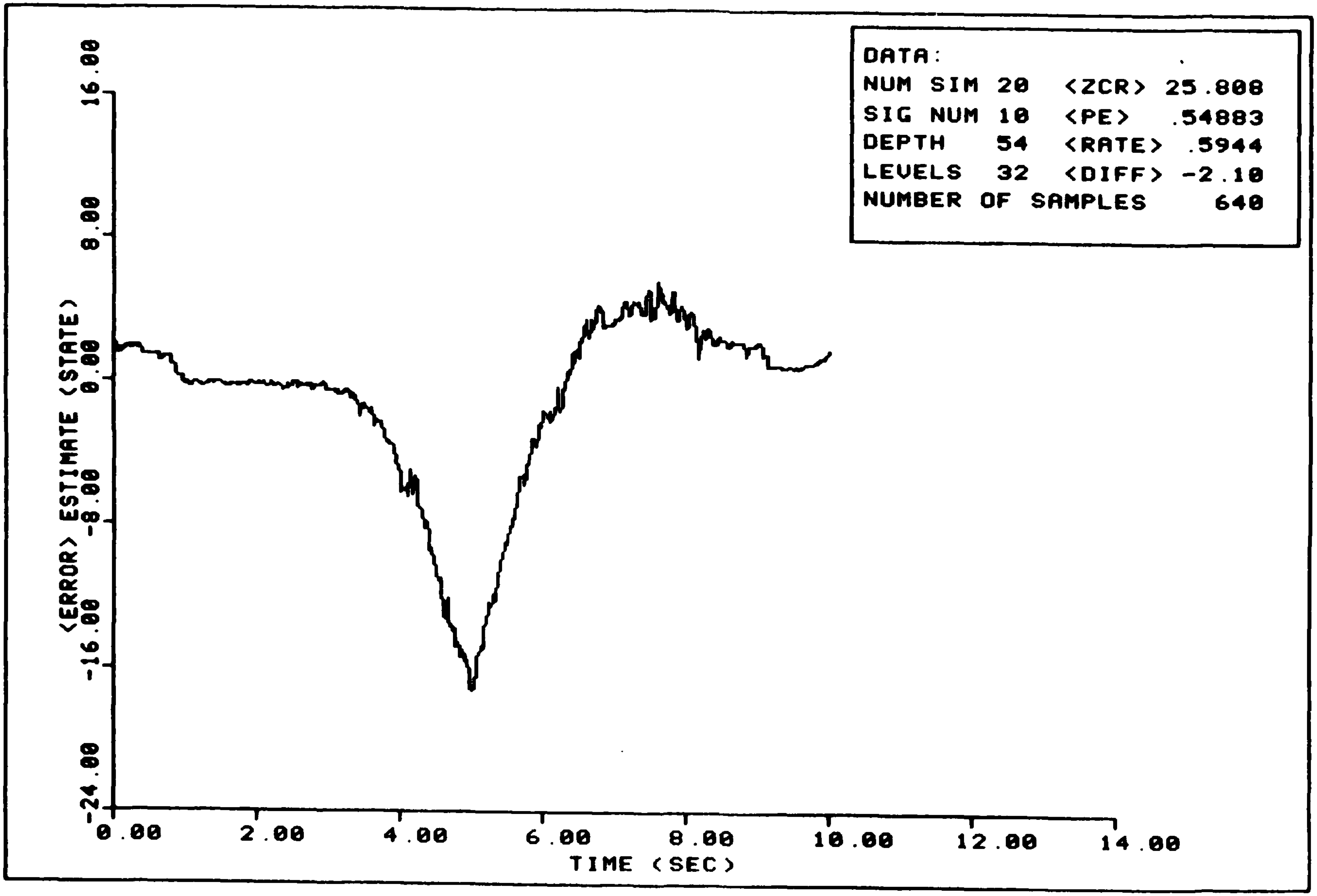


FIGURE 173. Ensemble Average State Error Performance

APPENDIX D

EXPANDED TRELLIS

EXPANDED TRELLIS:

The appendix presents the expanded trellis diagram used for the implementation of the Viterbi algorithm. The trellis of Figure 16 describes the possible state transitions of the Markov process for a given starting state. Connecting paths $\gamma(j, l)$ in Figure 16, exist only for state transitions for which $P\left\{X_j(k) \middle| X_l(K-1)\right\}$ is greater than zero. Since the branch metric of equation (53) is defined by taking the natural logrithm of the transition probability values, the zero probability of a given transition in the $\overline{P}$ matrix was replaced by:

$$P_{ij} = .000000000000001 \quad |i-j| > 1 \qquad (86)$$

the trellis of Figure 16 becomes the expanded trellis of Figure 174. Each state represents a possible pulse position. The solid lines represented the transitions the process may actually take. The dotted lines represent the transitions the process may not take, but now exist with an infintesimal probability of occurrence.

It is necessary to note that the "length" of the "illegal" paths is very long when compared to the legal paths, particularly under low $\langle ZCR \rangle$ conditions. Under high $\langle ZCR \rangle$, the difference in path length reduces as the trellis develops.

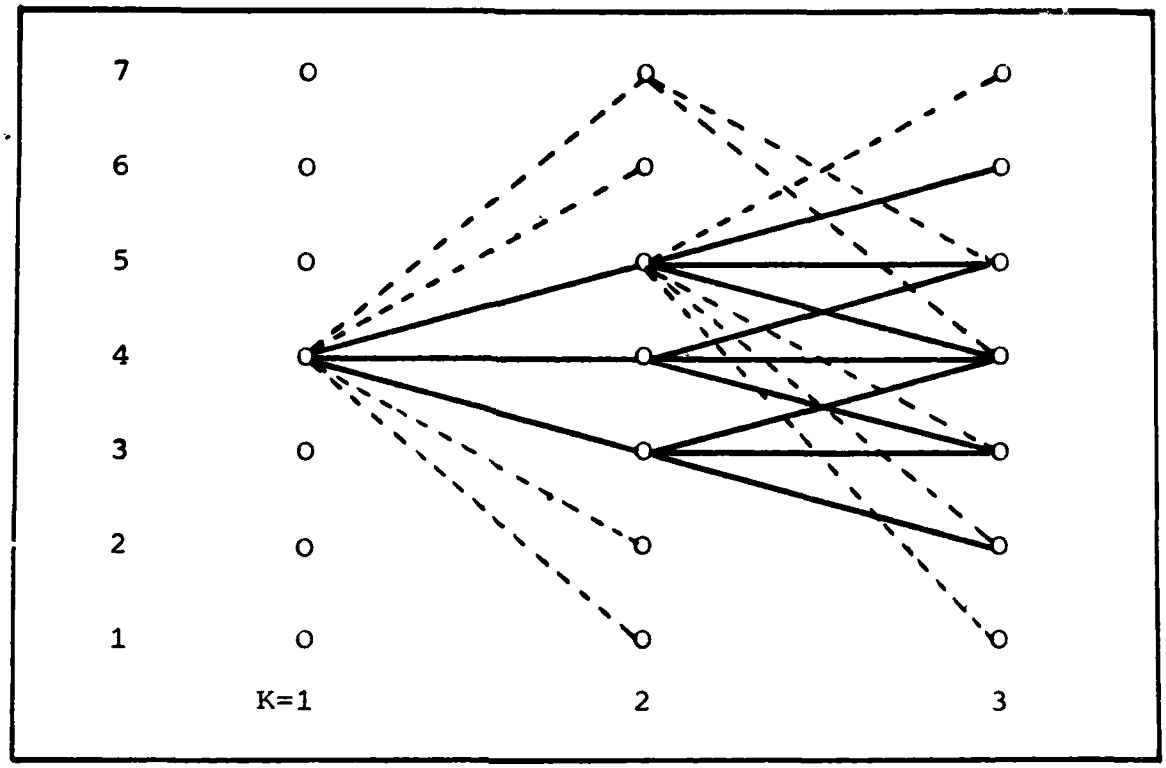


FIGURE 174. Expanded Trellis